The State of the Art in Visualizing Dynamic Graphs

Fabian Beck[1], Michael Burch[1], Stephan Diehl[2], and Daniel Weiskopf[1]

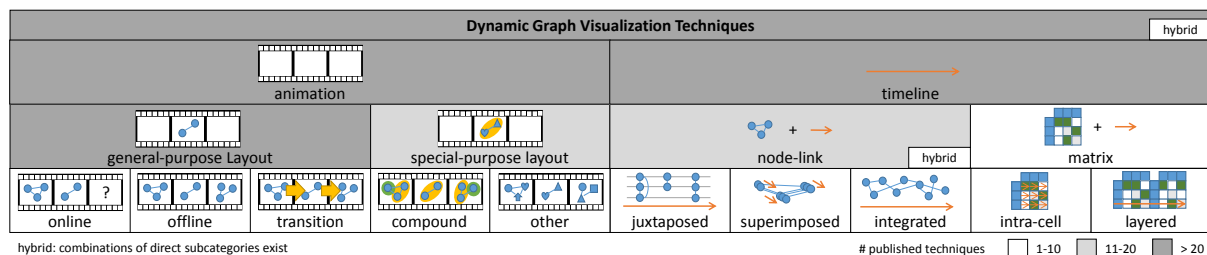[1]VISUS, University of Stuttgart, Germany
[2]University of Trier, Germany

Figure 1: *Illustrated hierarchical taxonomy of dynamic graph visualization techniques; the number of published techniques per taxonomic category is encoded in the brightness of the background (for details see Table 4).*

Abstract

Dynamic graph visualization focuses on the challenge of representing the evolution of relationships between entities in readable, scalable, and effective diagrams. This work surveys the growing number of approaches in this discipline. We derive a hierarchical taxonomy of techniques by systematically categorizing and tagging publications. While static graph visualizations are often divided into node-link and matrix representations, we identify the representation of time as the major distinguishing feature for dynamic graph visualizations: either graphs are represented as animated diagrams or as static charts based on a timeline. Evaluations of animated approaches focus on dynamic stability for preserving the viewer's mental map or, in general, compare animated diagrams to timeline-based ones. Finally, we identify and discuss challenges for future research.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI)

1. Introduction

The world is constantly evolving, there is nothing static or stable in it. But sometimes we pretend there is—just for simplification. In particular when analyzing data, this constraint is often applied: either we choose a single point in time or we aggregate longer spans of time. And indeed, the simplification is very helpful as it reduces the amount of data, makes computations faster, and simplifies reasoning as well as communication. However, this approach has its clear limitations: we learn nothing about the dynamics. As a consequence, we neither understand how and why certain stages are reached nor can foresee future changes.

Many aspects of the analog and digital world can be considered as objects being related to each other, for instance, people forming a social network, proteins interacting with each other, or components of a software system communicating through calls. We usually model relational data as *graphs* and a very active research community has formed around visualizing these structures: many visualization techniques have been introduced [vLKS*11], criteria for readable graph visualization have been studied [BRSG07]. And in fact, in most cases, the above simplification has been applied, visualizing *static* graphs only. However, over the years, researchers started to question this constraint and be-
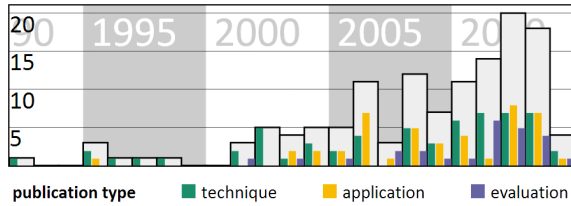
Figure 2: *Yearly number of publications on dynamic graph visualization according to our literature database; light gray bars indicate the total number of publications, colored bars distinguish the publications by type.*

Figure 3: *Different visual representations of static graphs as node-link or matrix diagrams all showing the same dataset.*

gan thinking about the visualization of *dynamic* graphs— relations between objects that change over time, as it is natural in the real world.

Starting in the 1990s with the problem of editing a static graph and visualizing the changes, the field was first understood as a subproblem of graph drawing: node-link diagrams need to be animated without destroying the user's mental image of the diagram, the so-called *mental map*. After the millennium, with the availability of more and more time-varying datasets, dynamic graph diagrams were discovered as an information visualization technique. Approaches became specialized to various application scenarios such as social network analysis or software engineering. Alternatives to animated node-link diagrams were introduced that plot the graph onto timelines. By 2010, the visualization of dynamic graphs was established as a standard visualization discipline. In consequence, the number of publications more than doubled from not more than 5 publications per year before 2006 to 20 publications in 2012 (Figure 2): evaluations were conducted comparing different techniques and exploring the role of the mental map, application areas were studied in greater detail, and still many new techniques and novel combinations of existing techniques were suggested. Visualizing dynamic graphs, hence, has become itself a very active and diverse research discipline involving several communities. What is missing so far, however, is a comprehensive survey of the area, structuring and discussing the variety of approaches and insights.

This paper is intended to fill this gap as it reports the state of the art in visualizing dynamic graphs. We give a brief introduction to the field (Section 2) and provide central definitions (Section 3). Based on a systematic literature search and categorization (Section 4), we build a hierarchical taxonomy of dynamic graph visualization and classify existing techniques into the taxonomy (Section 5); an illustrating representation of the taxonomy is provided in Figure 1. We also discuss evaluation results (Section 6) and applications for dynamic graph visualization approaches (Section 7). This systematic review finally allows us to identify challenges for future research (Section 8).
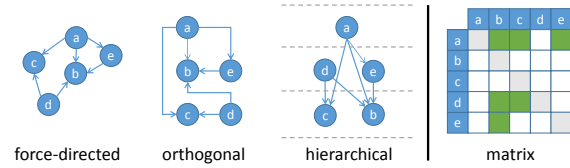
2. Background

Graphs in general form one of the most important data models in computer science because many problems and domains can be modeled as graph structures. Just to name a few, there are automata in theoretical computer science, flow networks such as pipes and roads, digital and non-digital social networks, computer networks such as the Internet, networks of companies and financial transactions, chemical reaction chains and molecular interactions, epidemic spreads of diseases in communities, or correlations of controlled variables in experiments. In most of those applications, temporal development can be observed and needs to be considered to fully understand the respective problem. Visualization is a particular means for exploratively comprehending and analyzing this data.

A graph consists of objects or entities, usually referred to as *vertices*, and relationships between them, called *edges*. Representing graphs as node-link diagrams, where vertices are drawn as visual nodes that are connected by graphical links representing the edges, has a long tradition. While the drawing first served illustration purposes only, gradually, layout algorithms were developed that allow one to automatically generate readable graph diagrams, for instance, *force-directed layouts*, which simulate physical forces between nodes, *orthogonal layouts*, where edges are plotted only along horizontal and vertical axes, or *hierarchical layouts*, which divide the graph into layers (Figure 3). As an independent field, *graph drawing* arose in the 1990s with the *Symposium on Graph Drawing*, which will be held in its 22nd edition in 2014. With an increasing interest in information visualization, also alternative visual representations of graphs have been introduced such as adjacency matrices (Figure 3). In such a matrix visualization, vertices are depicted as rows and columns of the matrix; colored cells of the matrix indicate whether two vertices are connected by an edge.

The characteristic difference of a *dynamic* graph to a *static* graph is that the structure of the vertices and edges as well as their attributes can change over time. Figure 4 shows an illustrating example of a dynamic graph and its visualization: a directed graph consisting of five nodes is visualized over three time steps as juxtaposed node-link diagrams. The position of the nodes is the same for all diagrams, which makes it easier to track the nodes over time. For instance, we see an
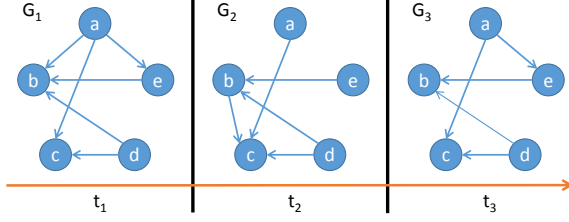
Figure 4: *Illustrating example of juxtaposed node-link diagrams on a timeline with constant node positions visualizing a directed graph with five vertices over three time steps.*

Table 1: *Examples of types of graphs that can be used for extending the basic definition of dynamic graphs.*

graph	$G_i :=$
(un)directed	(V_i, E_i) where $E_i \subseteq V_i \times V_i$ is either interpreted as directed or undirected
weighted	$(V_i, \overline{E}_i) : \overline{E}_i \subseteq V_i \times \mathbb{R}^+ \times V_i$
compound	$(V_i, E_i, E_i^T) : E_i^T$ hierarchy edges forming a tree
multivariate	$(V_i, \mathcal{E}_i, \rho) : \mathcal{E}_i = (\overline{E}_{i,1}, \ldots, \overline{E}_{i,k})$ list of sets of weighted edges and function ρ retrieves a multi-dimensional attribute vector for each $v \in V_i$

edge from node a to node e in the first time step (t_1), which disappears in the second (t_2) but reappears in the third (t_3). This small example should just give a first impression of a straightforward visualization of a dynamic graph—there exist much more sophisticated approaches as discussed in this survey.

Other publications have already partly reviewed the field of visualizing dynamic graphs. In 2001, Branke [Bra01] summarized the first animated node-link approaches "in a very early stage" of "dynamic and interactive graph drawing". In 2007, still focusing on animated node-link diagrams, Shannon and Quigley [SQ07] survey the field and conclude that "the issues unique to dynamic graphs are beginning to be uncovered in more depth". Since then, various user studies have been conducted on how important it is to preserve the mental map the user has in mind while watching animated node-link diagrams, summarized by Archambault and Purchase [AP13]. Further, Windhanger et al. [WZF11] discuss dynamic graph visualization from the application-specific perspective of organizational change in business networks. A brief review of dynamic graph visualization is also part of surveys of larger fields such as visualizing large graphs [vLKS*11] and force-directed layouts of node-link diagrams [Kob12]. Recently, Abello et al. [AAK*14] gave an overview of temporal multivariate graphs concentrating on node-link diagrams and surveying applications in software engineering in closer detail.

3. Dynamic Graph Data

Before starting to survey existing approaches, we need to clarify what data should be analyzed, what characteristics this data typically has, and what tasks are of interest for this data.

3.1. Data Model

To define a *dynamic graph*, we first introduce a *(static) graph* $G := (V, E)$, which models a set of objects V, called *vertices*, and their relationships $E \subseteq V \times V$, called *edges*. Then, a *dynamic graph* is defined as a sequence

$$\Gamma := (G_1, G_2, \ldots, G_n)$$

where $G_i := (V_i, E_i)$ are static graphs and indices refer to a sequence of time steps $\tau := (t_1, t_2, \ldots, t_n)$.

This basic definition can be interpreted and extended in different ways (Table 1 gives an overview): For instance, in a *directed* graph, $e_1 = (v, w)$ and $e_2 = (w, v)$ represent different relationships, whereas they are interpreted as the same relationship in an *undirected* graph. A *weighted* graph assigns a numeric attribute, called *weight*, to each edge. In graph theory, a *network* is a directed weighted graph, but overall, the term *network* is not used consistently in literature; for instance, a *social network* not always refers to a directed weighted graph. Further, a *compound* graph adds a hierarchical structure to the vertices, often used for interactively simplifying the graph by collapsing hierarchy vertices. The hierarchy can be considered as static over time as well as it might change together with the graph structure. In a *multivariate* dynamic graph, we have several attributes of edges or vertices that change over time. Moreover, application-specific extensions are possible but cannot be listed all. Of course, different extensions might be combined, for example, creating a *dynamic weighted directed compound graph*.

Please note that, similar to most of the approaches referenced in this survey, the above data model considers time as being *discrete*, but *ordinal* and *continuous* time scales [AMST11] can be represented indirectly: ordinal values could be mapped to virtual points on a discrete time scale; continuous processes that might form the basis of a dynamic graph need to be sampled to be represented in our data model. We also do not discern between *instants* and *intervals* [AMST11]: whether G_i is a snapshot at instant t_i or aggregates an interval around t_i. Often, it is not specified by the visualization technique which of the two models apply and rather depends on application domain and context. Abello et al. [AAK*14] discuss the modeling and representation of time for dynamic graphs in greater detail.

3.2. Graph Characteristics

The goal of analyzing graph data is to retrieve characteristic properties of its structure and attributes. These can be, for instance, topological properties that apply to the graph as a

whole, such as that the graph is *planar* (i.e., can be drawn as a 2D node-link diagram without any edge crossings), is *sparse* or *dense* (i.e., has few or many edges in relation to the possible number of edges), is *acyclic* (i.e., there is no cyclic path in a directed graph), is *bipartite* (i.e., the vertices fall into two sets, edges connecting only vertices from two different sets), etc. Also, individual properties of vertices and edges can be investigated such as retrieving all neighbors of a vertex, finding the shortest path between two vertices, or identifying clusters of vertices connected by many edges. These characteristics, however, only describe properties of static graphs. They are applicable to each graph individually in the sequence graphs, but there exist additional dynamic properties [BBD13]. A particularly important one is *dynamic variance*, which quantifies how much the graph structure changes from one time step to the next. Other properties are, for instance, whether the graph is only growing or shrinking, whether there are any trends in the evolution of vertex degrees or edge weights, or whether clusters or paths are preserved over time. Which of the properties are of interest, however, highly depends on the application the graph structure is used for. Typical graph analysis tasks are described elsewhere, for static graphs [LPP*06] and for specialized dynamic graphs [APS13, BPF14a].

4. Scope and Methodology

In order to retrieve a complete and structured list of references that forms the basis of this survey, we followed a systematic approach: we precisely defined the scope of the survey and retrieved relevant publications within that scope by manually searching through the relevant journals and conference proceedings as well as by following references of already retrieved publications. Through tagging, we then structured the retrieved literature.

4.1. Scope

The specific scope of this survey is visualizing dynamic graph structures as defined in Section 3.1. Some other visualization problems are specializations of the dynamic graph visualization problem or can be modeled as such, for instance, the visualization of static graphs, the comparison of two graphs, the visualization or comparison of hierarchies, or the visualization of time series. Although dynamic graph visualization techniques can be used to display such data, there are more specialized (usually, much more suitable) visualization techniques for these problems. Hence, we consider these specializations of the problem as out of scope for this survey. We also do not take into account approaches that first aggregate the dynamic graph (e.g., by using statistics or clustering) and then only visualize the simplified result, because the dynamic graph cannot be retrieved anymore from the displayed information. Moreover, there are theories and methods related to either dynamic graphs or visualization, which we cannot include into this survey: graph the-

ory, graph algorithms, visualization theory, interaction theory, perception, etc. We focus only on those aspects of those related fields that were directly applied to dynamic graph visualization.

4.2. Data Collection

Collecting the relevant publications for this survey, we started with a selection of papers that we knew from own previous research. We further manually scanned through all issues and proceedings of the main information visualization and graph drawing journals and conferences:

- **Journals**
  - *Computer Graphics Forum*
  - *IEEE Transactions on Visualization and Computer Graphics*
  - *Information Visualization*
  - *Journal of Graph Algorithms and Applications*
- **Conferences**
  - *IEEE Pacific Visualization Symposium (PacificVis)* [2001–2004: *InVis.au*; 2005–2007: *APVIS*]
  - *IEEE Symposium on Information Visualization (InfoVis)* [since 2006 a special issue of *IEEE Transactions on Visualization and Computer Graphics*]
  - *International Conference on Information Visualisation (IV)*
  - *Joint Eurographics–IEEE VGTC Symposium on Visualization (EuroVis)* [1999–2004: *VisSym*; since 2008 a special issue of *Computer Graphics Forum*]
  - *Symposium on Graph Drawing (GD)*

We followed citations in both directions: we checked the list of references in the paper to find older works and investigated citations of the paper using *Google Scholar*. Among those papers in scope of this survey, we only inserted by default peer-reviewed full papers published in journals and conferences written in English. If other criteria indicated certain impact and quality (e.g., high number of citations, remarkable contribution), we occasionally added papers not fulfilling all conditions.

4.3. Data Analysis

We applied tagging as the main instrument to structure the literature for this survey. Using tags instead of categorical dimensions provides the advantage that the publications can be assigned to multiple tags rather than just to one category per dimension. Categorical dimensions, however, better group the characteristics of an approach while tags usually are unstructured. Hence, we additionally defined tag categories (i.e., groups of tags belonging to the same dimension) for parts of the tags to also integrate this advantage into our tagging approach. In particular, we assigned a list of tags to each collected publication. We discussed the tags among the authors and defined the meaning of each tag in a short description. We further grouped important tags describing characteristics of similar kind into named categories. To systematically derive the list of tags and assign these to the publication, we used a process with three stages:

1. **Explorative Tagging:** We selected a small, arbitrary part of the collected publications and freely assigned reasonable tags. After analyzing some publications, we started to consolidate the tags by merging similar ones. Moreover, we built categories from tags describing the same dimension of characteristics. We continued with this tagging phase until we reached a stable list of central tags and categories.

2. **Category Tagging:** We then systematically classified all publications with respect to the tag categories. Each publication was assigned to at least one tag per category. Also some uncategorized or new tags were occasionally assigned during this process.

3. **Supplementary Tagging:** The final stage of the tagging was to analyze and compare groups of similar publications in detail. To systematically derive those groups sharing similar important characteristics, we restricted the bibliography by category tags assigned in the previous stage and combinations thereof. Group-specific supplementary tags were also identified.

4.4. Literature Dataset

The dataset we retrieved following the described methodology consists of 129 publications from 1992 to 2014. Five categories of tags have been identified in the tagging process: the general *type* of the publication, the visual representation of *time*, the visualization *paradigm* used for depicting the graph structure, the kind of *evaluation* performed, and the *application* addressed. We assigned at least one tag per category and usually a number of other tags to each publication. Figure 5 summarizes the result of the tagging process as a *tag cloud*, where the frequency of each tag is indicated by a subscript number and encoded in the font size. Additional to this quantified list of tags, Table 2 provides the descriptions of categories and included tags. In particular, the category tags formed the basis to derive a taxonomy of graph visualization techniques and to structure this survey. The complete dataset including all details and tags for each of the publication is available through an interactive Web tool[†].

The *type* of the publication forms one of the main features to discriminate the collected publications. *Technique* papers describing novel visualization approaches form the set of most important publications for this survey. All publications classified as such are described individually in this survey and classified into the taxonomy of visualization techniques (Section 5). We also systematically discuss all publications describing an *evaluation* of dynamic graph visualization techniques (Section 6). We further give an overview of *applications* of dynamic graph visualizations, however, not claiming completeness in this area (Section 7).

---

† http://go.visus.uni-stuttgart.de/dynamicgraphs

Table 2: *Categories and contained tags with descriptions.*

tag (category)	description
type	type of the paper
application	applying dynamic graph visualization to a specific application scenario
evaluation	empirical, algorithmic, or theoretical evaluation of visualization approaches
technique	novel visualization technique or system
time	visual representation of time
animation	mapping time to time in an animation
timeline	mapping time to space onto a timeline
generic	being applicable to all representations of time
paradigm	graph visualization paradigm
node-link	representing the graph as nodes connected by links
matrix	representing the graph as a visual adjacency matrix
generic	being applicable to all graph representations
evaluation	kind of evaluation
algorithmic	testing the presented approach algorithmically or using metrics
case_study	discussing a small number of application examples
expert	assessing the approach through external domain or visualization experts
none	no specific evaluation provided
survey	specially broad survey of related work
theoretical	theoretical considerations such as proof or runtime complexity
user_study	conducting a study involving other users
application	area of application
biology	bioinformatics data such as protein interactions or metabolic pathways
business	business- or economy-related data such as financial transactions, stock market, business processes
document	document collections, bibliometrics, and information retrieved from texts
eye_tracking	data recorded during eye-tracking experiments
infrastructure	infrastructure networks such as computer, communication, power, or road networks
media	data related to movies, TV, music, news and the like
social	social networks, social media, and other data from social life
software_engineering	information related software such as components, source code, developers, documentation, etc.
sports	sports-related data such as performance data or results
generic	no specific application suggested

5. Taxonomy and Classification of Dynamic Graph Visualization Techniques

Many different visualization techniques have been introduced for dynamic graph structures. In particular, we collected and classified 60 publications as technique papers. To provide a systematic overview of these techniques, we categorize the approaches according to a taxonomy. The taxonomy we developed for this purpose is structured hierarchically and consists of three layers, the first referenced with Roman numbers, the second with small letters in alphabetic order, and the third with Arabic numbers. While a first illustration of the taxonomy has been already presented in Figure 1, Table 4 provides a detailed description of its categories, their hierarchical structure, and the classification of techniques. This section describes all techniques and thereby follows as well the hierarchical structure of the taxonomy and employs the taxonomic categories as headlines. Additional to conceptional sketches of some of the presented approaches, small icons are used to illustrate and symbolize the categories of the taxonomy.
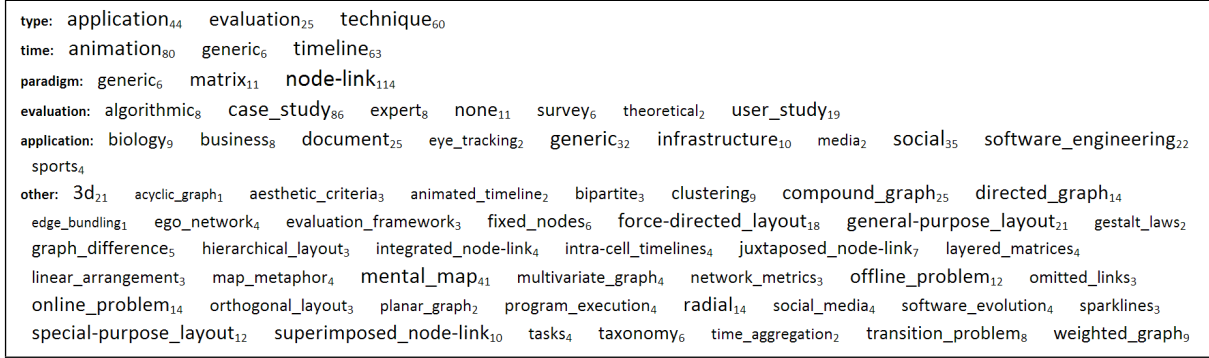
type: application$_{44}$ evaluation$_{25}$ technique$_{60}$

time: animation$_{80}$ generic$_6$ timeline$_{63}$

paradigm: generic$_6$ matrix$_{11}$ node-link$_{114}$

evaluation: algorithmic$_8$ case_study$_{86}$ expert$_8$ none$_{11}$ survey$_6$ theoretical$_2$ user_study$_{19}$

application: biology$_9$ business$_8$ document$_{25}$ eye_tracking$_2$ generic$_{32}$ infrastructure$_{10}$ media$_2$ social$_{35}$ software_engineering$_{22}$ sports$_4$

other: 3d$_{21}$ acyclic_graph$_1$ aesthetic_criteria$_3$ animated_timeline$_2$ bipartite$_3$ clustering$_9$ compound_graph$_{25}$ directed_graph$_{14}$ edge_bundling$_1$ ego_network$_4$ evaluation_framework$_3$ fixed_nodes$_6$ force-directed_layout$_{18}$ general-purpose_layout$_{21}$ gestalt_laws$_2$ graph_difference$_5$ hierarchical_layout$_3$ integrated_node-link$_4$ intra-cell_timelines$_4$ juxtaposed_node-link$_7$ layered_matrices$_4$ linear_arrangement$_3$ map_metaphor$_4$ mental_map$_{41}$ multivariate_graph$_4$ network_metrics$_3$ offline_problem$_{12}$ omitted_links$_3$ online_problem$_{14}$ orthogonal_layout$_3$ planar_graph$_2$ program_execution$_4$ radial$_{14}$ social_media$_4$ software_evolution$_4$ sparklines$_3$ special-purpose_layout$_{12}$ superimposed_node-link$_{10}$ tasks$_4$ taxonomy$_6$ time_aggregation$_2$ transition_problem$_8$ weighted_graph$_9$

Figure 5: *Tag cloud of manually assigned tags grouped by category summarizing all publications of our database; subscript numbers and font sizes refer to the usage frequency of the tags.*

Table 3: *Mapping between tags and taxonomy categories (■ required; □ optional).*

	I.	I.a.	I.a.1.	I.a.2.	I.a.3.	I.b.	I.b.1.	I.b.2.	II.	II.a.	II.a.1.	II.a.2	II.a.3.	II.a.4.	II.b.	II.b.1.	II.b.2.	III.
time																		
animation	■	■	■	■	■	■	■	■	·	·	·	·	·	·	·	·	·	■
timeline	·	·	·	·	·	·	·	·	■	■	■	■	■	■	■	■	■	■
paradigm																		
node-link	■	■	■	■	■	■	■	·	□	■	■	■	■	■	·	·	·	□
matrix	·	·	·	·	·	·	·	·	□	·	·	·	·	·	■	■	■	□
other																		
compound_graph	□	□	□	□	□	□	■	·	□	□	□	□	□	□	□	□	□	□
general-purpose_layout	□	■	■	■	■	·	·	·	□	□	□	□	□	□	□	□	□	□
integrated_node-link	·	·	·	·	·	·	·	·	□	□	·	·	■	□	·	·	·	□
intra-cell_timelines	·	·	·	·	·	·	·	·	□	·	·	·	·	·	□	■	·	□
juxtaposed_node-link	·	·	·	·	·	·	·	·	□	□	■	·	·	□	·	·	·	□
layered_matrices	·	·	·	·	·	·	·	·	□	·	·	·	·	·	□	·	■	□
offline_problem	□	□	·	■	□	□	□	□	·	·	·	·	·	·	·	·	·	□
online_problem	□	□	■	·	□	□	□	□	·	·	·	·	·	·	·	·	·	□
special-purpose_layout	□	·	·	·	■	■	■	·	·	·	·	·	·	·	·	·	·	□
superimposed_node-link	·	·	·	·	·	·	·	·	□	□	·	·	■	□	·	·	·	□
transition_problem	□	□	·	·	■	□	□	□	·	·	·	·	·	·	·	·	·	□

Please note that the taxonomy is pragmatically structuring existing techniques rather than exploring all possible concepts. Hence, combinations of concepts are not reflected if we have not found any example in the literature (e.g., a combination of animation and the matrix paradigm). This, however, only reflects the current state of the art and does not imply that a specific compound would be impossible or useless—the taxonomy might need to be extended through additional categories in the future. The specific mapping between tags and taxonomy categories can be retrieved from Table 3. The criteria to substructure the taxonomy are chosen from diverse categories of tags considering that a visualization technique cannot only be described through the employed visual mapping but as well through the requirements on data and algorithms. All combinations of tags can be explored through the provided literature database.

Two basic ways of visualizing a graph structure are *node-link diagrams* and *adjacency matrices*. As already illustrated in Figure 3, node-link diagrams represent vertices as graphical nodes that are connected by links; in a matrix, vertices are mapped to rows and columns of the matrix and a colored cell at an intersection of a row and column encodes an edge. While this would be one of the most important criteria to discern static graph visualizations, the time dimension adds another central aspect to the visualization when considering dynamic graphs. As Beck et al. [BBD09, BBD13] already discussed in this context, the time dimension can be mapped in an *animation* to a simulated time (time-to-time mapping) or to a space dimension of the generated visualization representing a *timeline* (time-to-space mapping). Other mappings would be possible—for instance, a mapping of time to color—but are rarely applied as an independent visualization approach. What can be found, however, are hybrid techniques that combine animation with timeline representations. Hence, the first level of the taxonomy divides the approaches into *animation*, *timeline*, and *hybrid* techniques.

I. Animation (Time-To-Time Mapping)
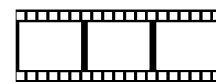
A mapping of the timestamps assigned to the sequence of graphs to visualization time results in an animated representation. Combining this straightforward map-

**Table 4:** *Hierarchical taxonomy of dynamic graph visualization techniques.*

Taxonomic Category	Illustration	#	Techniques
I. Animation (Time-to-Time Mapping)		33	
I.a. General-Purpose Layout		21	
I.a.1. Online Problem		6	layout adjustment to avoid overlap [MELS95] framework based on Bayesian decision theory [BW97] simulated annealing with customizable weights for optimization criteria [LLY06] efficient algorithm and GPU implementation [FT08b] consider age of nodes to stabilize the layout [GdBG12] more efficient initial positions of nodes [HMHU13]
I.a.2. Offline Problem		7	*Foresighted Layout (with Tolerance)* [DGK01, DG02, GBPD05] *GraphAEL*: force-directed layout with virtual forces between time steps [EHK*04b, FKN*05] *Visone*: force-directed layout with additional energy factors between time steps [BS08] user-selected multiple foci [FWSL12]
I.a.3. Transition Problem		8	stepwise animation for navigation based on a spring algorithm [HEW98] *Marey*: stepwise animation moving (parts of) the graph together [FE01, FE02, FH02, NF02] *VisuGraph*: using super-graph as intermediate step [LD08] transitions of bundled edges [HEF*13] *GraphDiaries*: highlight changes in staged transitions [BPF14a]
I.b. Special-Purpose Layout		12	
I.b.1. Compound Graphs		8	force-directed approach preserving the position of clusters [FT04] nested bubbles in 3D [KG06] *XLDN*: extending *Foresighted Layout with Tolerance* to dynamic compound graphs [PB08] focused animation collapsing constant parts of the hierarchy [RPD09] *ContexTour*: smooth contours of colored clusters [LSCL10] Space-filling maps of colored clusters [MKH12, HKV12] degree-of-interest functions for abstracting and focusing large graphs [AHSS13]
I.b.2. Other		4	online drawing of planar graphs [CDBT*92, CDBTT95] *DynaDAG*: acyclic graphs based on hierarchical layout [Nor96] stable layout of small world graphs [BFP06]
II. Timeline (Time-to-Space Mapping)		23	
II.a. Node-Link		15	
II.a.1. Juxtaposed		5	*TimeArcTrees*: linearized nodes on vertical axes [GBD09] *Parallel Edge Splatting*: artificially bipartite, linearized node layout [BVB*11, BBW12] nested circles: partial links in *TimeSpiderTrees* [BFBD10], ego centered graphs [FHQ11]
II.a.2. Superimposed		5	3D stack with fixed positions [BC03, DE02] 3D stack with relaxed positions [EKLN04, GW06] abstracting nodes and links to tubes [GHW09]
II.a.3. Integrated		3	ego network with edges as timelines [Rei10] ego network with ego node as timeline [SWW11] *Extended Massive Sequence Views*: event-based timeline with parallel edges [vdEHBvW13]
II.a.4. Hybrid (Juxt., Super., Int.)		2	juxtaposition as well as 2D and 3D superimposition [FAM*11, ITK10]
II.b. Matrix		8	
II.b.1. Intra-Cell Timelines		4	time series as sparkline bar charts [BSW13, YEL10] *Gestaltlines* encoding three metrics in angles and line lengths [BN11] pixel-based folded timelines [SWS10]
II.b.2. Layered Matrices		4	*(Layered) TimeRadarTrees*: radially layered lists with radial matrix thumbnails [BD08, BHW11] radially bended and layered matrices [VBSW13] *Cubix*: stacked matrices to a 3D cube and sliced small multiples thereof [BPF14b]
III. Hybrid (Animation, Timeline)		4	in situ integration of small visualizations [HSS11] cluster evolution on a timeline for navigating animated node-link diagrams [SMM12] moving timeline based on *Parallel Edge Splatting* [BBV*12] *DiffAni*: combinations of small multiples, difference representations, and animation [RM13]
		60	

ping with node-link diagrams creates a quite intuitive dynamic graph visualization: animated node-link diagrams. Up to now, still all animated approaches are based on node-link diagrams—we have not found any based on adjacency matrices. Hence, this taxonomy category refers to node-link approaches only. In nearly all approaches, the mental map is discussed. The term refers to the abstract structural information a user forms by looking at the layout of a graph. In the context of dynamic graph drawing, changes to this map should be minimal, in other words algorithms to draw sequences of graphs should preserve the mental map. To this end, the position of nodes is tried to be kept stable, which is called *dynamic stability*. This section is subdivided into *general-purpose layouts* and *special-purpose layouts* because having a specialized graph type, such as a compound graphs, changes the layout problem of node-link diagrams reasonably. Actually, specialized layouts such as for planar or acyclic graphs had been discussed before techniques for the drawing of general dynamic graphs were introduced.

I.a. General-Purpose Layout
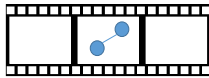
General-purpose layouts do not impose any requirements on the type of graph (cf. Table 1). They can be discerned, however, by whether they compute the individual node-link layouts of the animation only by considering past time steps (*online*) or both, past and future time steps (*offline*). In general, online approaches are more flexible as they are also applicable to scenarios where the complete evolution of the graph is not yet known when starting the animation (e.g., for interactively changed graphs or real-time monitoring). On the other hand, offline approaches allow for better optimizing the layout and maintaining the mental map because next changes are known. Other approaches are quite independent from the *online–offline problem*, but look in closer detail at the animated transition period between two consecutive layouts.
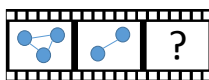
I.a.1. Online Problem The online layout problem first came up when interacting with static graphs: in particular, when showing only a subset of nodes and links or editing a graph, interactions lead to changes in the graph structure that should be displayed—hence, a sequence of graphs is visualized without knowing the full sequence from the beginning. Misue et al. [MELS95] introduce the first layout adaption approach for general graphs, which addresses the problem of graph editing and node overlap: in order to preserve the mental map, their force-directed approach maintains the original horizontal and vertical ordering of nodes while reducing overlap; the initial layout, however, is not computed by the approach. Brandes and Wagner [BW97] discuss an abstract framework based on Bayesian decision theory that describes the problem as a twofold model: a readability model of the individual graphs and a stability

model considering distances between the individual layouts and their predecessors. Basically, by multiplying quality factors from both models and optimizing the resulting functions, they derive a dynamic graph layout, which is implemented for a force-directed and an orthogonal approach. Lee et al. [LLY06] describe online layout as an optimization problem with customizable weights for different layout criteria; they apply simulated annealing for deriving a layout solution. Gorochowski et al. [GdBG12] suggest to use the concept of node age to preserve specifically the layout of old and stable graph structures. By introducing an efficient force-directed online layout algorithm and implementing it on the GPU, Frishman and Tal [FT08b] present a particularly fast layout approach. Also addressing efficiency, Hayashi et al. [HMHU13] investigate the effects of initial node placement on the responsiveness of a layout algorithm.
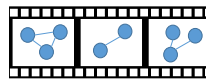
I.a.2. Offline Problem When a graph structure does not change through interactive editing or navigation, but through changes in the underlying domain, the full evolution is usually known at visualization time (an exception are monitoring systems). In this case, not only past but also future layouts can be considered for laying out the graphs of the time steps. This simplifies the layout problem and makes easier solutions applicable, the most straightforward one being to aggregate the full sequence of graphs and to lay out only this so-called *super-graph* (Figure 6)—individual layouts of the time steps are derived as a subset of the super-graph layout. However, adapting the layout gradually might be a better trade-off between preserving the mental map and individually readable layouts. Diehl et al. [DGK01] introduce *Foresighted Layout*, a generic framework that optimizes the straightforward super-graph approach: nodes are grouped if they are not active together and a super-graph is constructed from these grouped nodes applying an arbitrary static graph layout—due to grouping, node positions are reused if possible. Diehl and Görg [DG02] further extend this approach to *Foresighted Layout with Tolerance* adjusting the individual layouts derived from the super-graph within a certain level of deviation from the super-graph. The optimization of individual layouts can be realized with a force-directed algorithm [DG02], but as well with adapted algorithms for orthogonal and hierarchical layouts [GBPD05]. In *GraphAEL* [EHK*04b, FKN*05], the sequence of graphs is also aggregated but not into a super-graph: equivalent nodes are not merged but just connected through virtual edges. Considering theses edges and ignoring repulsive forces of nodes from different time steps, a single run of a force-directed algorithm determines the layout of all individual graphs. This technique can be applied to 2D and 3D animations [EHK*04b] or to hyperbolic and spherical spaces [FKN*05]. A similar approach was implemented for *Visone* [BS08] by introducing additional energy factors that increase with the position distance of equivalent nodes in adjacent time steps and need to be minimized.
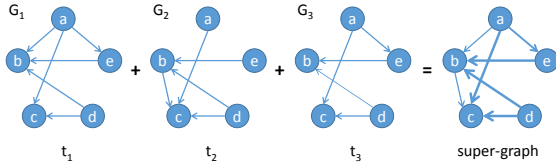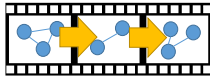
Figure 6: *Constructing a super-graph from a dynamic graph with three time-steps; edges occurring in multiple time steps are aggregated by higher edge weights (line thickness).*

Feng et al. [FWSL12] combine user-selected multiple foci with an offline approach: the focused nodes and their neighborhoods are enlarged.

**I.a.3. Transition Problem** Animating a node-link diagram does not only require to determine a sequence of layouts, but also the transitions between consecutive layouts need to be modeled—for the straightforward solution of just morphing one layout into the other, too many changes may happen at the same time to be traceable by the viewer. Huang et al. [HEW98] adapt a force-directed method for stepwise transitions when navigating a dynamic graph: when focusing a node, it is moved first to the center (with other nodes moving relative to it), then superfluous nodes disappear and new nodes appear, before finally the new layout is optimized. In *Marey* [FE01, FE02], the transition period consists of four phases: first, node and edges are removed if necessary; then, the graph is translated towards the new layout as if it is a single object; afterwards, the individual nodes are moved independently to their new positions; and finally, the new nodes and edges are shown. This approach is further extended by detecting clusters of nodes that share similar motions and moving theses together [FH02]; Nesbitt and Friedrich [NF02] suggest to use Gestalt laws to detect and structure the motion. In *VisuGraph* [LD08], a super-graph layout is computed with nodes in the same time step attracted to a specific position; for the transition between two time steps, the graph is first morphed to the super-graph to recalibrate the mental map, before the super-graph is further transformed into the layout of the next time-step. For graph layouts with fixed node positions, Hurter et al. [HEF*13] suggest an approach for transforming edge bundling smoothly between time steps. Bach et al. [BPF14a] use staged transitions similar to *Marey*, but specifically highlight removed and added elements; moreover, they use thumbnail images as previews for adjacent time steps.

I.b. Special-Purpose Layout

Specific characteristics of the graph may require, or at least profit from, other layout approaches than the ones presented for general dynamic graphs. All different

graph characteristics discussed in Section 3.2 could be considered in this context. Among these, dynamic compound graphs were addressed by many works while other characteristics have been investigated only occasionally yet. The current taxonomy category also includes online and offline approaches as well as techniques for transition between time steps; the special characteristics of the graph structure, however, discriminate the approaches more clearly.
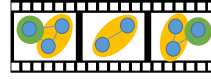
I.b.1. Compound Graph The additional hierarchy of a compound graph, which structures the set of nodes, can also be used to structure the visualization. In particular for larger graphs, it might help to abstract from single nodes to groups of nodes and can make the visualization more scalable. If approaches use clustering algorithms for creating a hierarchy or clusters, not only layout stability but also cluster stability needs to be optimized and clusters need to be tracked across time. Frishman and Tal [FT04] introduce an online approach for clustered graphs (i.e., compound graphs with only one level of clusters) based on a force-directed algorithm: boxes are drawn around the nodes of a cluster and the positions of clusters are tried to be preserved. Kumar and Garland [KG06] draw nested bubbles around nodes for a 3D graph layout to indicate the compound structure based on a force-directed offline approach. Pohl and Birke [PB08] extend *Foresighted Layout with Tolerance* [DG02] to compound graphs representing the hierarchy with nested boxes. Reitz et al. [RPD09] use the hierarchy to focus the animation and collapsing those hierarchies that stay unchanged with respect to the current time step. *ContexTour* [LSCL10] uses smooth contours with different colors to distinguish clusters, which creates a map-like image. Taking the map metaphor further, Mashima et al. [MKH12] and Hu et al. [HKV12] generate more space-filling drawings with directly bordering 'countries'. Abello et al. [AHSS13] discuss applying degree-of-interest functions to large graphs to highly aggregate parts of the graph while analyzing other parts in detail.

I.b.2. Other Dynamic graphs might have diverse characteristics that can be specifically considered for the layout. Actually, the publication that we regard as the first dynamic graph visualization according to the definitions and scope provided in Section 3.1 and Section 4.1 is specialized for variants of planar graphs: Cohen et al. published it as a technical report in 1992 [CDBT*92] and extended it as journal article in 1995 [CDBTT95]. They present a framework for drawing the graphs that warrants the planarity of the visual embedding. For acyclic graphs, *DynaDAG* [Nor96] extends the Sugiyama layout [STT81] for hierarchical drawings trying to preserve the mental map. Brandes et al. [BFP06] focus on drawing small-world graphs (i.e., graphs with short minimal distances between arbitrary nodes) and introduce a stable layout algorithm.
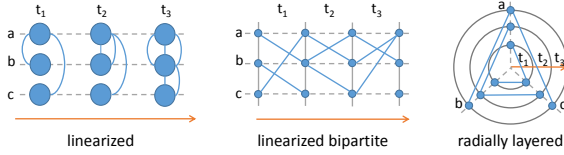
Figure 7: *Juxtaposed node-link approaches on a timeline.*

Figure 8: *Superimposed node-link approaches with different layers representing the time steps.*

II. Timeline (Time-to-Space Mapping)

Instead of using animation, the graph can be drawn onto a timeline in a time-to-space mapping. Timeline-based approaches promise to provide a better overview of time as they show the complete sequence of graphs in a static image [TMB02]. At least for small datasets, arbitrary points in time might be compared without interaction and characteristics of the graph could become traceable along the full evolution of the graph. However, only little space is available for drawing each of the graphs, which might decrease the readability of the diagram. This visual scalability problem is one of the main challenges for the techniques classified into this category. A main distinguishing feature of the approaches is whether they are based on node-link diagrams or on adjacency matrix representations.

II.a. Node-Link-Based Approaches

Placing node-link diagrams on a timeline is simple: as the introductory example in Figure 4 shows, node-link diagrams just need to be positioned next to each other, preferably applying a fixed layout of the nodes. In addition to this form of *juxtaposition*, other approaches can be used to lay out the diagrams: for instance, diagrams can be stacked on top of each other, which we call *superimposition*; or the sequence of graphs can be merged into an *integrated* diagram. These three categories—*juxtaposed*, *superimposed*, and *integrated*—are equivalent to the three categories Gleicher et al. [GAW*11] describe as generic approaches to visual comparison (while the first two categories carry the same name in this classification, *integration* is a form of *explicit encoding*). Similar categories are described by Javed and Elmqvist [JE12] with the main difference that *integrated* approaches are divided into *overloading* and *nesting*.
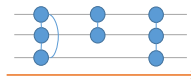
II.a.1. Juxtaposed

Juxtaposing node-link diagrams in a small multiples approach might be considered as simple, even trivial. However, considering this as a multiple views approach and just placing standard diagrams next to each other may not always produce convincing results: it is hard to see the differences between subsequent time steps, it is difficult to trace a node over several time steps, each diagram is qui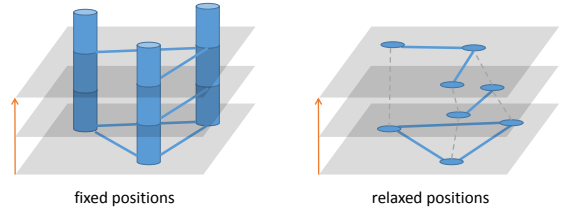te small, and the overall representation is likely cluttered even for small examples. Several approaches attempt to address these problems and suggest aligned and explicitly connected diagrams: *TimeArcTrees* [GBD09] arrange the nodes onto vertical axes and optimize the node ordering (Figure 7, linearized); this makes it easier to compare time steps and trace nodes, but visual scalability stays low. *Parallel Edge Splatting* [BVB*11] extends this approach by making each of the vertically arranged graphs artificially bipartite—edges are all directed from left to right (Figure 7, linearized bipartite). Additionally, plotting edge density instead of drawing overlapping links, this extension increases scalability. Also, a radial variant of *Parallel Edge Splatting* is possible [BBW12]. Other radially juxtaposed variants of node-link diagrams were introduced as well: *TimeSpiderTrees* [BFBD10] radially layer the node-link diagrams as nested circles (Figure 7, radially layered); but instead of completely drawing the links in those diagrams, only partial links are depicted (and expanded on demand). Farrugia et al. [FHQ11] use a similar layout to depict ego centered dynamic graphs (i.e., only neighboring nodes of a selected node are shown). Since ego networks are quite sparse and small, completely drawn links do not produce much clutter. To compare several ego networks, their radial representation can be juxtaposed as small multiples.
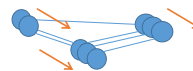
II.a.2. Superimposed

Instead of placing diagrams next to each other, they can be stacked on top of each other. In 2D, the nodes should have the same positions and the edges belonging to different time steps need to be discerned by color or stroke [EKLN04]. Stacked 2D diagrams can become 3D diagrams quite naturally through adding the time dimension as z-axis. However, as the third dimension is only used in discrete layers, these approaches are often referred to as 2.5D techniques. For instance, Dwyer and Eades [DE02] place 3D cylinders representing the nodes on an invisible 2D plane; edges at different levels indicate flows in the graph. Similarly, Brandes and Corman [BC03] depict nodes as cylinders, but also add transparent planes that help discern the stacked layers (Figure 8, fixed positions). Erten et al. [EKLN04], in contrast, allow the same node to have different positions on the layers; to preserve the mental map, they use an adapted force-directed layout algorithm that
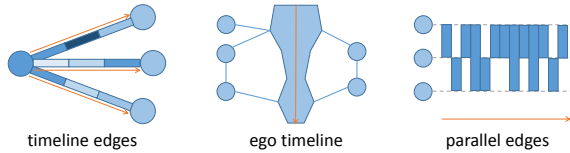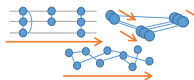
Figure 9: *Node-link approaches with an integrated timeline.*

Figure 10: *Different visualization approaches for intra-cell timelines in matrix representations.*

moves the same nodes to at least similar positions across layers (Figure 8, relaxed positions). Groh et al. [GHW09] extend this approach by abstracting from nodes and links: they just use the connections between the same nodes of different layers (i.e., the dashed lines in Figure 8, relaxed positions) and visualize these as 3D tubes.

II.a.3. Integrated In an integrated diagram, the timeline is woven into the node-link diagram—diagrams for different time steps cannot be separated anymore without destroying the layout. For instance, Reitz [Rei10] shows ego networks where the evolution of edge weights is plotted onto each edge by varying its color in sections; each edge, hence, forms an individual timeline (Figure 9, edge timelines). Shi et al. [SWW11] focus on ego networks as well and introduce an approach they call *1.5D layout*: they connect a central timeline representing the ego node with its neighbors by a link at the point in time where an edge first appears (Figure 9, ego timeline). For graphs where edges represent instant events (they can be assigned to a specific point in time without having a duration), van den Elzen et al. [vdEHBvW13] suggest *Extended Massive Sequence Views*, a technique that arranges a list of events as a horizontal timeline and orders nodes onto the vertical axis; edges are drawn at the respective position of the timeline as vertical lines connecting two vertical node positions (Figure 9, parallel edges). The authors discuss various strategies to linearly arrange the nodes as well as a radial variant of the visualization. In general, integrated approaches seem to be often restricted to only special types of dynamic graphs. An advantage is that the integration of the timeline allows the representation of arbitrarily fine samplings of continuous time.

II.a.4. Hybrid (Juxtaposed, Superimposed, Integrated) The different approaches to map node-link diagrams onto a timeline can also be combined. A simple approach is to have different views for the approaches and to smoothly transform one into the other, for instance, juxtaposed node-link diagrams as well as 2D and 3D superimposed ones [FAM*11, ITK10]. While Federico et al. [FAM*11] suggest three predefined views (camera perspective and layer positions), Itoh et al. [ITK10] allow the user to independently set camera perspective and layer positions.
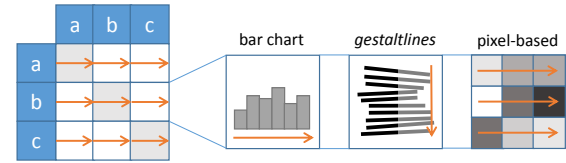
II.b. Matrix-Based Approaches

Since adjacency matrices are used for visualizing static graphs, they can also be employed for encoding dynamic graphs on a timeline. The challenge is to connect the spatial encoding of time with the matrix information, which seems to be harder than in node-link diagrams because matrices are less flexible with respect to layout. However, advantages of the matrix representations—for instance, staying more readable for larger and denser graphs [GFC05, KEC06]—justify to tackle these difficulties. Depending on how the timeline and matrix are combined, we identify two types of visualization techniques based on adjacency matrices.

II.b.1. Intra-Cell Timelines As a dimensional stacking approach, the cells of an adjacency matrix may each contain an individual timeline to represent the dynamic changes of the edge encoded in the particular cell; the small intra-cell representations are a form of time series encoded in a *sparkline* [Tuf06]. As illustrated in Figure 10, very different forms of intra-cell timeline representations exist. For instance, it is possible to embed a simple bar chart showing the time on a horizontal axis from left to right [BSW13, YEL10] (Figure 10, left). In particular, Burch et al. [BSW13] show how a hierarchy structuring the vertices can be attached to sides of the matrix to represent a dynamic compound graph. Yi et al. [YEL10] extend the basic approach by displaying aggregated timelines for the vertices and encoding different edge types with different colors. When the cells become too small, the timeline representations become colored cells (as when showing a static graph). Instead of bars, Brandes and Nick [BN11] use so-called *Gestaltlines* as intra-cell representations: stacked lines encode three metrics, one in their angle, one in their length to the left, and one in their length to the right (Figure 10, middle). Moreover, Stein et al. [SWS10] suggest a pixel-based approach that folds a timeline into a cell so that each pixel (or any other quadratic subdivision) of the cell represents a point in time (Figure 10, right). The weight of the edge at a specific point is encoded in the brightness of the pixel; different folding strategies such as *row-by-row*, *column-by-column*, *diagonal*, etc. are possible.
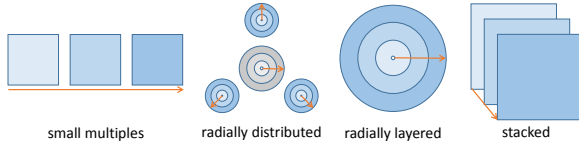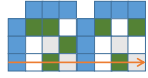
Figure 11: *Schematic illustrations of layered matrices approaches; three shades of blue symbolize three time steps.*

Figure 12: *Distribution of time representations and graph visualization paradigms among evaluation (left) and application publications (right).*

II.b.2. Layered Matrices Instead of splitting the cells of a matrix, adjacency matrices can be juxtaposed or layered on a timeline. While the straightforward approach is to use adjacency matrices as small multiples (Figure 11, small multiples), there are a couple of more sophisticated approaches: The *TimeRadarTrees* approach [BD08, BHW11] radially layers the time steps encoding the list of edges. Details can be read from radial thumbnail images attached to each circle segment representing a vertex; these thumbnails form a kind of distributed matrix representation (Figure 11, radially distributed). A radially layered approach by Vehlow et al. [VBSW13] literally bends the matrices of each time step into rings of a circle (Figure 11, radially layered). *Cubix* [BPF14b] stacks 2D matrices into 3D (Figure 11, stacked) and provides different small-multiples representations where adjacency matrices or other slices of the 3D matrix are juxtaposed.

III. Hybrid (Animation, Timeline)
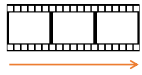
While most dynamic graph visualization techniques can be unambiguously classified as either using animation or using a static timeline, a few approaches combine both mappings of time. The combination of the two time representations can, however, follow different strategies in those hybrid approaches. We consider as hybrid ones only those strategies that use both representations closely connected and cannot easily be split into independent techniques. Hadlak et al. [HSS11] suggest small in situ visualizations to be integrated into a larger visualization; in that way also animated diagrams can be embedded into a timeline. Sallaberry et al. [SMM12] use a timeline-based, aggregated representation of cluster evolution to navigate through an animated node-link diagram. Beck et al. [BBV*12] animate long sequences of graphs as a moving timeline representation based on *Parallel Edge Splatting*. *DiffAni* [RM13] allows the user, for instance, to interactively aggregate parts of a timeline representation into animations; the authors present a taxonomy of hybrid approaches that can be systematically constructed from small multiples, difference representations, and animation—*DiffAni* supports arbitrary combinations.

6. Evaluation

Most papers we collected contain some sort of evaluation (Figure 5, category *evaluation*). Nevertheless, most evaluations are 'only' case studies, which are a rather lightweight form of evaluation not necessarily involving users (86 out of 129 publications). In contrast, some papers specifically focus on evaluation (25 publications, Figure 5, category *type*). This section primarily discusses insights gained from these evaluation papers, but also reflect some interesting evaluation results from technique or application papers. The section is structured according to different types of evaluation (Table 2 and Figure 5, category *evaluation*). Since surveys on the field have been already discussed in Section 2, they are omitted here. Also case studies and expert reviews being lightweight evaluation techniques are not discussed in further detail. Figure 12 (left) shows that most evaluation approaches focus on animated node-link diagrams, but only few on timeline representations; matrix visualizations are not yet evaluated in the context of dynamic graphs.

6.1. Evaluation Frameworks

Visualizations are explorative analysis tools and often do not address a single task, but families of tasks. For evaluating visualization approaches, however, it should be clarified which tasks are addressed in the evaluation; a task taxonomy can help selecting appropriate tasks. Extending a task taxonomy for static graphs [LPP*06], the taxonomy by Ahn et al. [APS13] collects and structures tasks by three dimensions: *entity* (granularity such as nodes and links, groups, or complete network), *property* (topology of entities and domain-specific attributes), and *temporal feature* (states over time); these dimensions and their subcategories span a design space of tasks. Bach et al. [BPF14a] introduce an alternative taxonomy dividing tasks into temporal tasks (*when*), topological tasks (*where*), and behavioral tasks (*what*). A further tool for evaluation is identifying desired properties of the visualization, which are often called *aesthetic criteria* in context of graph visualization. While many of those criteria were discussed and tested for static graphs [BRSG07], Beck et al. [BBD09, BBD13] extend these criteria to dynamic graph visualization and suggest three dimensions grouping the criteria: *general aesthetic criteria*, *dynamic aesthetic criteria*, and *aesthetic scalability criteria*. These criteria dimensions can be used for evaluating qualities of dynamic graph visualizations and find the right visualization technique for a given set of tasks and datasets [BBD13].

6.2. Algorithmic Evaluation

A rather technical method of evaluation is measuring characteristics of the layout algorithm. For instance, Frishman and Tal [FT04] compare runtime, node density, and cluster characteristics of their online approach for drawing compound graphs to other approaches—the results indicate that their approach better discerns clusters. For another online approach, Frishman and Tal [FT08b] follow a similar evaluation scheme and provide evidence that their approach maintains the mental map well and its implementation on the GPU is more than a magnitude faster than a CPU implementation. The same approach is applied by Gorochowski et al. [GdBG12], however, combining it with a visual analysis of the sample data because not all important features are covered by the employed set of metrics. Other approaches as well measure runtime performance [FWSL12], preservation of the mental map [CTB13, FWSL12, HMHU13], or cluster quality [LSCL10]. While evaluation constitutes only a part of the above papers, Brandes and Mader [BM12] focus only on an algorithmic evaluation of offline node-link approaches: they contrast layout stability to individual layout quality and conclude that only linking nodes of adjacent time steps provides better results than positioning nodes at or near fixed positions.

6.3. User Studies

While algorithmic evaluation provides first insights into the characteristics of a visualization, finally deciding whether a technique is helpful requires involving users. In controlled experiments, different parameters of a visualization technique can be tested or two approaches might be compared against each other under controlled conditions. We group the studies according to research question they investigate.

Mental Map: The role of the mental map has been discussed since the first works on dynamic graph visualization and is probably their best evaluated aspect. While we briefly summarize results of related studies, Archambault and Purchase [AP13] review studies on the mental map in much greater detail. In a first study, Purchase et al. [PHG07] test different degrees of preserving the mental map for a hierarchical node-link layout and find a positive effect of mental map preservation for some of the tasks. However, in similar studies, Purchase and Samra [PS08] and Saffrey and Purchase [SP08] cannot confirm positive effects of preserving the mental map but, in contrast, find that favoring a good individual layout tends to produce better results. Further, Archambault and Purchase [AP12] do not detect significant effects of preserving the mental map. Ghani et al. [GEY12] vary the layout between fixed positions (perfect mental map) and individually optimized layouts (no mental map); their results, however, show that the mental map condition performs better. Hence, although several studies have been conducted so far, the role of the mental map is not yet clear, but at least

there are indications that its role might have been overestimated in the literature.

Animation vs. Timeline: A central question that is reflected by the two main categories of our taxonomy is the representation of time—are animated approaches better than timeline-based ones, or vice versa? For a graph with time-varying node attributes visualized as a node-link diagram with static positions, Saraiya et al. [SLN05] compare an animated slider solution to an approach with small time-series visualizations in the nodes: they observe better performance of participants for the animated approach when only one or two points in time need to be studied, but better results for the timeline approach for tasks involving more time steps. Farrugia and Quigley [FQ11] contrast an animated node-link diagram to a static approach showing node-link diagrams in a grid (timeline) based on the same node layout; for the investigated time-related tasks, the static approach generally tends to provide better performance with respect to error rates and response time. With a similar experiment design, Archambault et al. [APP11a] also find generally quicker response times for the timeline conditions; but for some tasks related to the appearance of entities, animation produces lower error rates. In a qualitative study, Boyandin et al. [BBL12] further show that animation tends to reveal more findings on adjacent time steps while small multiples foster the discovery of patterns lasting over longer periods. In conclusion of these studies, timeline-based approaches seem to be preferable for tasks involving more than two time steps; for other tasks, the studies do not yet provide a clear picture. As Rufiange and McGuffin [RM13] show, hybrid approaches mixing animation and timeline, under certain conditions, can produce better results.

Specific Approaches: Further, some specific approaches were evaluated, either varying a visualization parameter or comparing two approaches against each other. Elmqvist and Tsigas [ET03] show that their timeline-based node-link approach works better than Hasse diagrams for visualizing the information flow between interacting software processes with respect to most tasks. Rey and Diehl [RD10] investigate animation speed and labeling in animated node-link diagrams and find that an interactively selectable presentation speed does not have a positive effect on comprehension performance, but that always showing labels instead of retrieving labels only on demand is beneficial. Archambault et al. [APP11b] test the importance of explicitly encoding differences in node-link diagrams for two animated approaches and a small multiples approach; difference maps help for certain comparison-related tasks and are preferred by the participants. For another difference visualization approach, Zaman et al. [ZKS11] also find conditions where it outperforms animation. Bach et al. [BPF14a] compare their animated node-link approach *GraphDiaries* to simpler approaches without staged-animation or without animation at all; their results provide some evidence that animation is helpful and staged-animation tends to further improve task performance.

7. Application

Dynamic graph visualizations can be applied to various datasets from very different domains. In this section, we present in particular those publications classified as application papers, but occasionally also reference technique papers that put a special focus on a certain application. Despite the variety of application scenarios, only few mature tools are readily available for dynamic graph visualization: both *Gephi* [BHJ09] and *Commetrix* [Tri06] show animated node-link diagrams and provide features for filtering, clustering, and computing network metrics. Selecting only the applications papers in our database, we see in Figure 12 (right) that almost only node-link diagrams are used, but in combination with both animation and timeline representations.

Social Data: With the increasing popularity of social media, a variety of datasets of social networks became available and a new area of research formed around this topic. A challenge in this context is to visualize the dynamics and evolution of these networks. Moody et al. [MMBd05] motivate the use of dynamic graph visualization for social network analysis and provide initial examples of how animated node-link approaches can be used for visualizing these networks. Bender-deMoll and McFarland [BdM06] present a framework for testing different animated node-link layouts with respect to this application. Social network metrics can be used to specialize and augment graph layouts [CTB13]. Brandes et al. [BIM12] discuss the visualization of analysis models for temporal effects in social networks. Specific types of dynamic social networks that have been investigated are, for example, textual online conversations such as chats [PT10], online communities [ATMS*11], the activity and interest of bloggers [IYTK12], the development of character relationships in literature [IA12, OKK13], or the propagation of microblogging messages [LQC*13]. Studying the evolution of these networks might support answering sociological and psychological questions, could help historians and literature scholars, but may as well act as an end-user tool to retrieve facts from social media, for instance, to find relevant news or people.

Documents: Texts and documents can be related through different kinds of connections such as citations, hyperlinks, similar content, etc. If documents are created or existing ones are changed, these connections change over time. For researchers, quite natural applications are libraries of scientific publications, for instance, visualizing co-authorship [EHK*04a, Rei10, YAPM08], connections between research areas [EHK*04a], co-citation networks [Che06] (i.e., publications cited by the same other publication), or semantic similarity of content [ABPdO12]. But, of course, the evolution of other document collections can be visualized as well, such as hyperlink structures between webpages [TK05, YAPM08], semantic relationships between retrieved entities [SNF10], or between messages in news streams [GHN13]. Since documents have a social

context (authors, readers, distributors, etc.), there is certain overlap with social data—whether people or entities form the nodes of the dynamic graph may act as a distinguishing criterion.

Software Engineering: Among the first, dynamic graph visualizations were applied to software-related data. Already in 1995, Kimelman et al. [KLRZ95] discuss the problem of visual complexity for visualizing program executions. Also visualizing the execution dynamics of software, others depict the information flow between processes [ET03], object interactions [GLW06], dynamic call graphs [BMR*12, BBV*12], or objects migrating between hosts [FT08a]. In addition to execution, another dynamic aspect of software that can be visualized is its evolution—the changes applied to software systems over time: for instance, the evolution of call, inheritance, or flow graphs [CKN*03], of co-changed files [BH06], or author–file relationships [OM08, OM09]. Hence, software systems have different dimensions of time—execution and evolution—and relevant graph data can be derived and combined from multiple sources.

Others: There are various further applications of dynamic graphs and their visualization. In research, for instance in context of biology, evolving metabolic pathways [RUK*10], simulated chemical reaction networks [JSS*12] and uncertainties therein [VHK*13], or protein interaction networks [BFL12] are studied. Psychology and user interface research may profit from depicting eye gaze data as dynamic graphs recorded in eye-tracking studies [BBR*14, HEF*13]. Computer scientists can investigate the evolution of the Internet [BBP08] or anomalies in communication networks [LSW13]. Business researchers and mangers are supported in analyzing contagion in financial networks [vLDBF13] and movements in stock portfolios [DE02]. Dynamic graph drawing may even reach a wider audience when depicting popular topics such as movie–actor affiliations [BHP06] or international soccer matches [AFH*10]. In essence, every application where static graph visualization can be applied is a candidate for also leveraging the visualization of dynamic graphs—static graphs are often just a simplification of dynamic ones.

8. Research Challenges

The overview of the field forms a good basis for discussing the main challenges of, and possible directions for future research. Our tagging of publications and taxonomy of techniques allow us to see which areas have already been studied in detail and which areas are just covered by few works; the survey of evaluation results identifies answered and unanswered questions; the discussed applications point towards areas where dynamic graph visualization might have a considerable practical impact. Specifically for the visualization of multivariate dynamic graphs, Abello et al. [AAK*14] already discussed open problems, partly overlapping with the research challenges introduced in the following.

Evaluation: A first, quite generic research challenge is conducting evaluations. As in other areas of information visualization, only a few questions have been evaluated so far. Although most efforts have concentrated on the importance of the mental map, it is still unclear when and to what extent the mental map needs to be preserved. Only a few other questions have been investigated. Hence, it is still unanswered which visualization technique is suitable for which application. Also, setting the parameters of the different techniques is rarely explored, such as magic constants of layout and clustering algorithms, color-mapping, or sampling rate and time aggregation. Moreover, questions of perception are relevant, for instance, the users' cognitive load watching an animation, effects of attention, or typical misinterpretations of visual elements.

Visual Scalability: A further general challenge in information visualization, which also applies here, is visual scalability: with an increasing amount of displayed data, the generated diagrams should remain readable. In most of the discussed applications, datasets should not be limited to a few dozens of vertices. For static diagrams, already some scalability comparisons have been conducted, for instance, contrasting node-link and matrix diagrams [GFC05, KEC06]. While, in static graphs, the number of nodes and the density of edges are the two crucial parameters for scalability, Beck et al. [BBD09] argue that the number of time steps needs to be considered as a third dimension when studying dynamic graph visualizations. So far, scalability has only played a minor role in designing most dynamic graph visualization approaches, although it has been already discussed in depth for static graphs [vLKS*11].

Hybrids: Some approaches already started to combine different techniques, such as animation and timeline representations (Taxonomy Category III.) or different variants of timeline-based node-link diagrams (Taxonomy Category II.a.4.). However, many more possible hybrids exist between different categories of the taxonomy. While not all combinations are sensible, there are a number of hybrid variants that appear to be promising: for instance, the combination of node-link diagrams and matrices have been already successfully explored for static graphs [HF07, HFM07]. Animating a matrix diagram should also work if the animation process is designed with similar care as in those works describing the transition problem for node-link diagrams (Taxonomy Category I.a.3.). Further, combinations of animation and timeline approaches have not yet been fully explored.

Extended Data Dimensions: While dynamic graphs add a time dimension to static graphs, dynamic graphs themselves can be extended by other dimensions. As it is the case when we move from static to dynamic graphs, adding a new data dimension usually requires the visualization to change considerably. What has been already studied, is adding a hierarchy to the vertices of the graph (Taxonomy Category I.b.1. and partly II.a.1.). Yet, most of these approaches assume that the hierarchy is constant or at least only gradually changing. An open question is how to visualize a hierarchical structure that changes more significantly along with the dynamic graph. Additional data dimensions that have only been partly explored are dynamic multivariate graphs [BN11, YEL10, AAK*14], dynamic graphs with uncertainty information [VHK*13], and geo-located graphs [HEF*13]. Finally, the effects of using continuous time with arbitrary fine sampling rates, rather than discretized time, are largely unexplored.

Interaction: While most timeline-based approaches produce static images, animated node-link diagrams inherently require interactive displays. Although the first works in the field already investigated smooth navigation in graphs, this focus seems to have been lost over the years. Navigation in dynamic graphs can have multiple dimensions: users might navigate in space (i.e., the static graph) as well as in time. As recent works with a special focus on navigation have shown, interaction could help integrate different modes [RM13, HSS11] or filter down the data to a manageable size [AHSS13]. Working with several views introduces further challenges, such as, visually mapping and synchronizing multiple representatives of the same objects between different visualizations. Finally, annotating and editing a dynamic graph structure is not well studied. All these interactive features and combinations of different views and visualizations can be integrated into consistent visual analytics approaches.

Applications: Without application, visualization would lose its purpose. As discussed in Section 7, there are a number of areas where different dynamic graph visualizations have been already applied. Looking at the recent development, we see that still new applications have arisen, for instance, the analysis of characters in literature [IA12, OKK13], understanding financial and bio-medical contagion networks [vLDBF13], or visualizing eye-tracking data [BBR*14, HEF*13]. This steady progress already suggests that there are still more areas to be explored for dynamic graph visualization. In particular, looking at the wide variety of applications that static graphs have, we will probably find many examples where dynamics are important but have not been studied by means of visualization yet. As datasets often have application-specific characteristics and there exist application-specific requirements, an adaption of existing techniques might not be straightforward but could require new research.

9. Conclusions

This work presented the state of the art in visualizing dynamically changing relational data. Building on previous work in the graph drawing and information visualization communities, the visualization of dynamic graphs has become an active and constantly growing research discipline. While, initially, animated node-link diagrams dominated the re-

search, timeline-based approaches gained more importance recently. First empirical evaluations of the approaches were conducted. With the increasing availability of temporal data, dynamic graph visualizations have been adapted to many areas of application. By systematically collecting and categorizing the literature, we structured the field and collected the works spread over different areas of research and communities. This review makes approaches more comparable as it points out similarities and differences using a consistent taxonomy. Studying the development of the field based on the collected data, there are no indicators that research within the area has met an end. On the contrary, new visualizations are still introduced, evaluation is only in its infancy, and there are many open challenges and interesting research problems.

References

[AAK*14] ABELLO J., ARCHAMBAULT D., KENNEDY J., KOBOUROV S. G., MA K.-L., MIKSCH S., MUELDER C., TELEA A.: Temporal multivariate networks. In *Multivariate Network Visualization*. Springer, 2014, ch. 8, pp. 151–175. 3, 14, 15

[ABPdO12] ALENCAR A. B., BÖRNER K., PAULOVICH F. V., DE OLIVEIRA M. C. F.: Time-aware visualization of document collections. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (2012), SAC, ACM, pp. 997–1004. doi:10.1145/2245276.2245469. 14

[AFH*10] AHMED A., FU X., HONG S.-H., NGUYEN Q. H., XU K.: Visual analysis of history of World Cup: A dynamic network with dynamic hierarchy and geographic clustering. In *Visual Information Communication*, VINCI. Springer, 2010, pp. 25–39. doi:10.1007/978-1-4419-0312-9_2. 14

[AHSS13] ABELLO J., HADLAK S., SCHUMANN H., SCHULZ H.-J.: A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2013), 337–350. doi:10.1109/TVCG.2013.109. 7, 9, 15

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of time-oriented data*. Springer, 2011. doi:10.1007/978-0-85729-079-3. 3

[AP12] ARCHAMBAULT D., PURCHASE H. C.: The mental map and memorability in dynamic graphs. In *Proceeding of the IEEE Pacific Visualization Symposium* (2012), PacificVis, IEEE, pp. 89–96. doi:10.1109/PacificVis.2012.6183578. 13

[AP13] ARCHAMBAULT D., PURCHASE H. C.: The "Map" in the mental map: Experimental results in dynamic graph drawing. *International Journal of Human-Computer Studies 71*, 11 (2013), 1044 – 1055. doi:10.1016/j.ijhcs.2013.08.004. 3, 13

[APP11a] ARCHAMBAULT D., PURCHASE H., PINAUD B.: Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics 17*, 4 (2011), 539–552. doi:10.1109/tvcg.2010.78. 13

[APP11b] ARCHAMBAULT D., PURCHASE H. C., PINAUD B.: Difference map readability for dynamic graphs. In *Graph Drawing*, GD. Springer, 2011, pp. 50–61. doi:10.1007/978-3-642-18469-7_5. 13

[APS13] AHN J.-w., PLAISANT C., SHNEIDERMAN B.: A task taxonomy for network evolution analysis. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2013), 365–376. doi:10.1109/TVCG.2013.238. 4, 12

[ATMS*11] AHN J.-w., TAIEB-MAIMON M., SOPAN A., PLAISANT C., SHNEIDERMAN B.: Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, SBP. Springer, 2011, pp. 309–316. doi:10.1007/978-3-642-19656-0_43. 14

[BBD09] BECK F., BURCH M., DIEHL S.: Towards an aesthetic dimensions framework for dynamic graph visualisations. In *Proceedings of the 13th International Conference on Information Visualisation* (2009), IV, IEEE, pp. 592–597. doi:10.1109/IV.2009.42. 6, 12, 15

[BBD13] BECK F., BURCH M., DIEHL S.: Matching application requirements with dynamic graph visualization profiles. In *Proceedings of the 17th International Conference on Information Visualisation* (2013), IV, IEEE, pp. 11–18. doi:10.1109/IV.2013.2. 4, 6, 12

[BBL12] BOYANDIN I., BERTINI E., LALANNE D.: A qualitative study on the exploration of temporal changes in flow maps with animation and small-multiples. *Computer Graphics Forum 31*, 3pt2 (2012), 1005–1014. doi:10.1111/j.1467-8659.2012.03093.x. 13

[BBP08] BOITMANIS K., BRANDES U., PICH C.: Visualizing internet evolution on the autonomous systems level. In *Graph Drawing* (2008), GD, Springer, pp. 365–376. doi:10.1007/978-3-540-77537-9_36. 14

[BBR*14] BURCH M., BECK F., RASCHKE M., BLASCHECK T., WEISKOPF D.: A dynamic graph visualization perspective on eye movement data. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (2014), ETRA, pp. 151–158. doi:10.1145/2578153.2578175. 14, 15

[BBV*12] BECK F., BURCH M., VEHLOW C., DIEHL S., WEISKOPF D.: Rapid serial visual presentation in dynamic graph visualization. In *Proceedings of the 2012 IEEE Symposium on Visual Languages and Human-Centric Computing* (2012), VL/HCC, IEEE, pp. 185–192. doi:10.1109/vlhcc.2012.6344514. 7, 12, 14

[BBW12] BURCH M., BECK F., WEISKOPF D.: Radial Edge Splatting for visualizing dynamic directed graphs. In *Proceedings of the 4th International Conference on Information Visualization Theory and Applications* (2012), IVAPP, SciTePress, pp. 603–612. 7, 10

[BC03] BRANDES U., CORMAN S. R.: Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization 2*, 1 (2003), 40–50. doi:10.1057/palgrave.ivs.9500037. 7, 10

[BD08] BURCH M., DIEHL S.: TimeRadarTrees: Visualizing dynamic compound digraphs. *Computer Graphics Forum 27*, 3 (2008), 823–830. doi:10.1111/j.1467-8659.2008.01213.x. 7, 12

[BdM06] BENDER-DEMOLL S., MCFARLAND D. A.: The art and science of dynamic network visualization. *Journal of Social Structure 7*, 2 (2006), 1–38. URL: http://www.cmu.edu/joss/content/articles/volume7/deMollMcFarland/. 14

[BFBD10] BURCH M., FRITZ M., BECK F., DIEHL S.: TimeSpiderTrees: A novel visual metaphor for dynamic compound graphs. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (2010), VL/HCC, IEEE, pp. 168–175. doi:10.1109/vlhcc.2010.31. 7, 10

[BFL12] BRASCH S., FUELLEN G., LINSEN L.: VENLO: Interactive visual exploration of aligned biological networks and their evolution. In *Visualization in Medicine and Life Sciences II*. Springer, 2012, pp. 229–247. doi:10.1007/978-3-642-21608-4_13. 14

[BFP06] BRANDES U., FLEISCHER D., PUPPE T.: Dynamic spectral layout of small worlds. In *Graph Drawing* (2006), GD, Springer, pp. 25–36. doi:10.1007/11618058_3. 7, 9

[BH06] BEYER D., HASSAN A. E.: Animated visualization of software history using Evolution Storyboards. In *Proceedings of the 13th Working Conference on Reverse Engineering* (2006), WCRE, IEEE Computer Society, pp. 199–210. doi:10.1109/wcre.2006.14. 14

[BHJ09] BASTIAN M., HEYMANN S., JACOMY M.: Gephi: an open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media* (2009), ICWSM, The AAAI Press, pp. 361–362. URL: http://aaai.org/ocs/index.php/ICWSM/09/paper/view/154. 14

[BHP06] BRANDES U., HOEFER M., PICH C.: Affiliation dynamics with an application to movie-actor biographies. In *Proceedings of the 8th Joint Eurographics / IEEE VGTC Conference on Visualization* (2006), EuroVis, Eurographics Association, pp. 179–186. doi:10.2312/VisSym/EuroVis06/179-186. 14

[BHW11] BURCH M., HÖFERLIN M., WEISKOPF D.: Layered TimeRadarTrees. In *Proceedings of the 15th International Conference on Information Visualisation* (2011), IV, IEEE, pp. 18–25. doi:10.1109/IV.2011.93. 7, 14

[BIM12] BRANDES U., INDLEKOFER N., MADER M.: Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. *Social Networks 34*, 3 (2012), 291–308. doi:10.1016/j.socnet.2011.06.002. 14

[BM12] BRANDES U., MADER M.: A quantitative comparison of stress-minimization approaches for offline dynamic graph drawing. In *Graph Drawing* (2012), GD, Springer, pp. 99–110. doi:10.1007/978-3-642-25878-7_11. 13

[BMR*12] BURCH M., MÜLLER C., REINA G., SCHMAUDER H., GREIS M., WEISKOPF D.: Visualizing dynamic call graphs. In *Vision, Modeling & Visualization* (2012), VMV, The Eurographics Association, pp. 207–214. doi:10.2312/PE/VMV/VMV12/207-214. 14

[BN11] BRANDES U., NICK B.: Asymmetric relations in longitudinal social networks. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2283–2290. doi:10.1109/TVCG.2011.169. 7, 11, 15

[BPF14a] BACH B., PIETRIGA E., FEKETE J.-D.: GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (2014), 740–754. doi:10.1109/TVCG.2013.254. 4, 7, 9, 12, 13

[BPF14b] BACH B., PIETRIGA E., FEKETE J.-D.: Visualizing dynamic networks with matrix cubes. In *Proceedings of the SICCHI Conference on Human Factors in Computing Systems* (2014), CHI. doi:10.1145/2556288.2557010. 7, 12

[Bra01] BRANKE J.: Dynamic graph drawing. In *Drawing graphs*. Springer, 2001, pp. 228–246. doi:10.1007/3-540-44969-8_9. 3

[BRSG07] BENNETT C., RYALL J., SPALTEHOLZ L., GOOCH A.: The aesthetics of graph visualization. In *Computational Aesthetics in Graphics, Visualization, and Imaging* (2007), pp. 57–64. doi:10.2312/COMPAESTH/COMPAESTH07/057-064. 1, 12

[BS08] BAUR M., SCHANK T.: *Dynamic Graph Drawing in Visone*. Tech. rep., Fakultät für Informatik, Universität Karlsruhe, 2008. URL: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000007990. 7, 8

[BSW13] BURCH M., SCHMIDT B., WEISKOPF D.: A matrix-based visualization for exploring dynamic compound digraphs. In *Proceedings of the 17th International Conference Information Visualisation* (2013), IV, IEEE, pp. 66–73. doi:10.1109/IV.2013.8. 7, 11

[BVB*11] BURCH M., VEHLOW C., BECK F., DIEHL S., WEISKOPF D.: Parallel Edge Splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2344–2353. doi:10.1109/tvcg.2011.226. 7, 10

[BW97] BRANDES U., WAGNER D.: A Bayesian paradigm for dynamic graph layout. In *Graph Drawing* (1997), GD, Springer, pp. 236–247. doi:10.1007/3-540-63938-1_66. 7, 8

[CDBT*92] COHEN R. F., DI BATTISTA G., TAMASSIA R., TOLLIS I. G., BERTOLAZZI P.: A framework for dynamic graph drawing. In *Proceedings of the 8th Annual Symposium on Computational Geometry* (1992), SoCG, ACM, pp. 261–270. doi:10.1145/142675.142728. 7, 9

[CDBTT95] COHEN R. F., DI BATTISTA G., TAMASSIA R., TOLLIS I. G.: Dynamic graph drawings: Trees, series-parallel digraphs, and st-digraphs. *SIAM Journal on Computing 24*, 5 (1995), 970–1001. doi:10.1137/S0097539792235724. 7, 9

[Che06] CHEN C.: CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology 57*, 3 (2006), 359–377. doi:10.1002/asi.20317. 14

[CKN*03] COLLBERG C., KOBOUROV S., NAGRA J., PITTS J., WAMPLER K.: A system for graph-based visualization of the evolution of software. In *Proceedings of the 2003 ACM Symposium on Software Visualization* (2003), SoftVis, ACM, pp. 77–ff. doi:10.1145/774833.774844. 14

[CTB13] CENGIZ TURKER U., BALCISOY S.: A visualization technique for large temporal social network datasets in hyperbolic space. *Journal of Visual Languages & Computing 25*, 3 (2013), 227–242. doi:10.1016/j.jvlc.2013.10.008. 13, 14

[DE02] DWYER T., EADES P.: Visualising a fund manager flow graph with columns and worms. In *Proceedings of the 6th International Conference on Information Visualisation* (2002), IV, IEEE, pp. 147–152. doi:10.1109/IV.2002.1028770. 7, 10, 14

[DG02] DIEHL S., GÖRG C.: Graphs, they are changing – dynamic graph drawing for a sequence of graphs. In *Graph Drawing*, GD. Springer-Verlag, 2002, pp. 23–31. doi:10.1007/3-540-36151-0_3. 7, 8, 9

[DGK01] DIEHL S., GÖRG C., KERREN A.: Preserving the mental map using foresighted layout. In *Proceedings of the 3rd Joint Eurographics–IEEE TCVG Conference on Visualization* (2001), VisSym, Eurographics Association, pp. 175–184. doi:10.2312/VisSym/VisSym01/175-184. 7, 8

[EHK*04a] ERTEN C., HARDING P. J., KOBOUROV S. G., WAMPLER K., YEE G.: Exploring the computing literature using temporal graph visualization. In *Visualization and Data Analysis* (2004), VDA, pp. 45–56. doi:10.1117/12.539245. 14

[EHK*04b] ERTEN C., HARDING P. J., KOBOUROV S. G., WAMPLER K., YEE G.: GraphAEL: Graph animations

with evolving layouts graph drawing. In *Graph Drawing*, GD. Springer, 2004, pp. 98–110. `doi:10.1007/978-3-540-24595-7_9`. 7, 8

[EKLN04] ERTEN C., KOBOUROV S. G., LE V., NAVABI A.: Simultaneous graph drawing: Layout algorithms and visualization schemes. In *Graph Drawing* (2004), GD, Springer, pp. 437–449. `doi:10.1007/978-3-540-24595-7_41`. 7, 10

[ET03] ELMQVIST N., TSIGAS P.: Causality visualization using animated growing polygons. In *Proceedings of the 2003 IEEE Symposium on Information Visualization* (2003), InfoVis, IEEE, pp. 189–196. `doi:10.1109/INFVIS.2003.1249025`. 13, 14

[FAM*11] FEDERICO P., AIGNER W., MIKSCH S., WINDHAGER F., ZENK L.: A visual analytics approach to dynamic social networks. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies* (2011), i-KNOW, p. 47. `doi:10.1145/2024288.2024344`. 7, 11

[FE01] FRIEDRICH C., EADES P.: The Marey graph animation tool demo. In *Graph Drawing* (2001), GD, Springer, pp. 396–406. `doi:10.1007/3-540-44541-2_37`. 7, 9

[FE02] FRIEDRICH C., EADES P.: Graph drawing in motion. *Journal of Graph Algorithms and Applications 6*, 3 (2002), 353–370. `doi:10.7155/jgaa.00057`. 7, 9

[FH02] FRIEDRICH C., HOULE M. E.: Graph drawing in motion II. In *Graph Drawing* (2002), GD, Springer, pp. 220–231. `doi:10.1007/3-540-45848-4_18`. 7, 9

[FHQ11] FARRUGIA M., HURLEY N., QUIGLEY A.: Exploring temporal ego networks using small multiples and tree-ring layouts. In *Proceedings of the 4th International Conference on Advances in Computer-Human Interactions* (2011), ACHI, pp. 79–88. URL: `http://hcibib.org/ACHI11`. 7, 10

[FKN*05] FORRESTER D., KOBOUROV S. G., NAVABI A., WAMPLER K., YEE G. V.: graphael: A system for generalized force-directed layouts. In *Graph drawing* (2005), GD, Springer, pp. 454–464. `doi:10.1007/978-3-540-31843-9_47`. 7, 8

[FQ11] FARRUGIA M., QUIGLEY A.: Effective temporal graph layout: A comparative study of animation versus static display methods. *Information Visualization 10*, 1 (2011), 47–64. `doi:10.1057/ivs.2010.10`. 13

[FT04] FRISHMAN Y., TAL A.: Dynamic drawing of clustered graphs. In *Proceedings of the 2004 IEEE Symposium on Information Visualization* (2004), InfoVis, IEEE, pp. 191–198. `doi:10.1109/INFVIS.2004.18`. 7, 9, 13

[FT08a] FRISHMAN Y., TAL A.: MOVIS: A system for visualizing distributed mobile object environments. *Journal of Visual Languages & Computing 19*, 3 (2008), 303–320. `doi:10.1016/j.jvlc.2007.07.006`. 14

[FT08b] FRISHMAN Y., TAL A.: Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 727–740. `doi:10.1109/tvcg.2008.11`. 7, 8, 13

[FWSL12] FENG K.-C., WANG C., SHEN H.-W., LEE T.-Y.: Coherent time-varying graph drawing with multifocus+context interaction. *IEEE Transactions on Visualization and Computer Graphics 18*, 8 (2012), 1330–1342. `doi:10.1109/TVCG.2011.128`. 7, 9, 13

[GAW*11] GLEICHER M., ALBERS D., WALKER R., JUSUFI I., HANSEN C. D., ROBERTS J. C.: Visual comparison for information visualization. *Information Visualization 10*, 4 (2011), 289–309. `doi:10.1177/1473871611416549`. 10

[GBD09] GREILICH M., BURCH M., DIEHL S.: Visualizing the evolution of compound digraphs with TimeArcTrees. *Computer Graphics Forum 28*, 3 (2009), 975–982. `doi:10.1111/j.1467-8659.2009.01451.x`. 7, 10

[GBPD05] GÖRG C., BIRKE P., POHL M., DIEHL S.: Dynamic graph drawing of sequences of orthogonal and hierarchical graphs. In *Graph Drawing* (2005), GD, Springer, pp. 228–238. `doi:10.1007/978-3-540-31843-9_24`. 7, 8

[GdBG12] GOROCHOWSKI T. E., DI BERNARDO M., GRIERSON C. S.: Using aging to visually uncover evolutionary processes on networks. *IEEE Transactions on Visualization and Computer Graphics 18*, 8 (2012), 1343–1352. `doi:10.1109/TVCG.2011.142`. 7, 8, 13

[GEY12] GHANI S., ELMQVIST N., YI J. S.: Perception of animated node-link diagrams for dynamic graphs. *Computer Graphics Forum 31*, 3pt3 (2012), 1205–1214. `doi:10.1111/j.1467-8659.2012.03113.x`. 13

[GFC05] GHONIEM M., FEKETE J.-D., CASTAGLIOLA P.: On the readability of graphs using node-link and matrix-based representations: A controlled experiment and statistical analysis. *Information Visualization 4*, 2 (2005), 114–135. `doi:10.1057/palgrave.ivs.9500092`. 11, 15

[GHN13] GANSNER E. R., HU Y., NORTH S.: Interactive visualization of streaming text data with dynamic maps. *Journal of Graph Algorithms and Applications 17*, 4 (2013), 515–540. `doi:10.7155/jgaa.00302`. 14

[GHW09] GROH G., HANSTEIN H., WÖRNDL W.: Interactively visualizing dynamic social networks with DySoN. In *Proceedings of the 2009 Workshop on Visual Interfaces to the Social and the Semantic Web* (2009), VISSW. URL: `http://data.semanticweb.org/workshop/VISSW/2009/paper/main/10/html`. 7, 11

[GLW06] GREEVY O., LANZA M., WYSSEIER C.: Visualizing live software systems in 3D. In *Proceedings of the 2006 ACM Symposium on Software Visualization* (2006), SoftVis, ACM, pp. 47–56. `doi:10.1145/1148493.1148501`. 14

[GW06] GAERTLER M., WAGNER D.: A hybrid model for drawing dynamic and evolving graphs. In *Graph Drawing* (2006), GD, Springer, pp. 189–200. `doi:10.1007/11618058_18`. 7

[HEF*13] HURTER C., ERSOY O., FABRIKANT S., KLEIN T., TELEA A.: Bundled visualization of dynamic graph and trail data. *IEEE Transactions on Visualization and Computer Graphics* (2013). `doi:10.1109/TVCG.2013.246`. 7, 9, 14, 15

[HEW98] HUANG M. L., EADES P., WANG J.: On-line animated visualization of huge graphs using a modified spring algorithm. *Journal of Visual Languages & Computing 9*, 6 (1998), 623–645. `doi:10.1006/jvlc.1998.0094`. 7, 9

[HF07] HENRY N., FEKETE J.-D.: MatLink: Enhanced matrix visualization for analyzing social networks. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-Computer Interaction* (2007), INTERACT, Springer, pp. 288–302. `doi:10.1007/978-3-540-74800-7_24`. 15

[HFM07] HENRY N., FEKETE J. D., MCGUFFIN M. J.: NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1302–1309. `doi:10.1109/tvcg.2007.70582`. 15

[HKV12] HU Y., KOBOUROV S. G., VEERAMONI S.: Embedding, clustering and coloring for dynamic maps. In *Proceedings of the 2012 IEEE Pacific Visualization Symposium* (2012), PacificVis, IEEE Computer Society, pp. 33–40. `doi:10.1109/PacificVis.2012.6183571`. 7, 9

[HMHU13] HAYASHI A., MATSUBAYASHI T., HOSHIDE T., UCHIYAMA T.: Initial positioning method for online and real-time dynamic graph drawing of time varying data. In *Proceedings of the 17th International Conference on Information Visualisation* (2013), IV, pp. 435–444. `doi:10.1109/IV.2013.57`. 7, 8, 13

[HSS11] HADLAK S., SCHULZ H.-J., SCHUMANN H.: In situ exploration of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2334–2343. `doi:10.1109/TVCG.2011.213`. 7, 12, 15

[IA12] ITOH M., AKAISHI M.: Visualizing for changes in relationships between historical figures in chronicles. In *Proceedings of the 16th International Conference on Information Visualisation* (2012), IV, pp. 283–290. `doi:10.1109/IV.2012.55`. 14, 15

[ITK10] ITOH M., TOYODA M., KITSUREGAWA M.: An interactive visualization framework for time-series of web graphs in a 3D environment. In *Proceedings of the 14th International Conference on Information Visualisation* (2010), IV, IEEE, pp. 54–60. `doi:10.1109/IV.2010.18`. 7, 11

[IYTK12] ITOH M., YOSHINAGA N., TOYODA M., KITSUREGAWA M.: Analysis and visualization of temporal changes in bloggers' activities and interests. In *Proceeding of the 2012 IEEE Pacific Visualization Symposium* (2012), PacificVis, IEEE, pp. 57–64. `doi:10.1109/PacificVis.2012.6183574`. 14

[JE12] JAVED W., ELMQVIST N.: Exploring the design space of composite visualization. In *Proceeding of the IEEE Pacific Visualization Symposium* (2012), PacificVis, IEEE, pp. 1–8. `doi:10.1109/PacificVis.2012.6183556`. 10

[JSS*12] JOHN M., SCHULZ H.-J., SCHUMANN H., UHRMACHER A. M., UNGER A.: Constructing and visualizing chemical reaction networks from pi-calculus models. In *Formal Aspects of Computing*, vol. 25. Springer, 2012, pp. 723–742. `doi:10.1007/s00165-011-0209-0`. 14

[KEC06] KELLER R., ECKERT C. M., CLARKSON P. J.: Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization 5*, 1 (2006), 62–76. `doi:10.1057/palgrave.ivs.9500116`. 11, 15

[KG06] KUMAR G., GARLAND M.: Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 805–812. `doi:10.1109/tvcg.2006.193`. 7, 9

[KLRZ95] KIMELMAN D., LEBAN B., ROTH T., ZERNIK D.: Reduction of visual complexity in dynamic graphs. In *Graph Drawing* (1995), GD, Springer, pp. 218–225. `doi:10.1007/3-540-58950-3_373`. 14

[Kob12] KOBOUROV S. G.: Spring embedders and force directed graph drawing algorithms. *arXiv preprint arXiv:1201.3011* (2012). URL: `http://arxiv.org/abs/1201.3011`. 3

[LD08] LOUBIER E., DOUSSET B.: Temporal and relational data representation by graph morphing. *Safety and Reliability for Managing Risk 14*, 02 (2008), 2008–16. 7, 9

[LLY06] LEE Y.-Y., LIN C.-C., YEN H.-C.: Mental map preserving graph drawing using simulated annealing. In *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation* (2006), APVis, Australian Computer Society, Inc., pp. 179–188. 7, 8

[LPP*06] LEE B., PLAISANT C., PARR C. S., FEKETE J. D., HENRY N.: Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors*

(New York, NY, USA, 2006), BELIV, ACM, pp. 1–5. `doi:10.1145/1168149.1168168`. 4, 12

[LQC*13] LI Q., QU H., CHEN L., WANG R., YONG J., SI D.: Visual analysis of retweeting propagation network in a microblogging platform. In *Proceedings of the 6th International Symposium on Visual Information Communication and Interaction* (2013), VINCI, ACM, pp. 44–53. `doi:10.1145/2493102.2493108`. 14

[LSCL10] LIN Y.-R., SUN J., CAO N., LIU S.: ContexTour: Contextual contour visual analysis on dynamic multi-relational clustering. In *Proceedings of the 2010 SIAM International Conference on Data Mining* (2010), SDM, pp. 418–429. `doi:10.1137/1.9781611972801.37`. 7, 9, 13

[LSW13] LIAO Q., SHI L., WANG C.: Visual analysis of large-scale network anomalies. *IBM Journal of Research and Development 57*, 3/4 (2013), 13:1–13:12. `doi:10.1147/JRD.2013.2249356`. 14

[MELS95] MISUE K., EADES P., LAI W., SUGIYAMA K.: Layout adjustment and the mental map. *Journal of Visual Languages & Computing 6*, 2 (1995), 183–210. `doi:10.1006/jvlc.1995.1010`. 7, 8

[MKH12] MASHIMA D., KOBOUROV S., HU Y.: Visualizing dynamic data with maps. *IEEE Transactions on Visualization and Computer Graphics 18*, 9 (2012), 1424–1437. `doi:10.1109/TVCG.2011.288`. 7, 9

[MMBd05] MOODY J., MCFARLAND D., BENDER-DEMOLL S.: Dynamic network visualisation. *American Journal of Sociology 110*, 4 (2005), 1206–1241. `doi:10.1086/421509`. 14

[NF02] NESBITT K. V., FRIEDRICH C.: Applying gestalt principles to animated visualizations of network data. In *Proceedings of the 6th International Conference on Information Visualisation* (2002), IV, IEEE, pp. 737–743. `doi:10.1109/IV.2002.1028859`. 7, 9

[Nor96] NORTH S. C.: Incremental layout in DynaDAG. In *Graph Drawing* (1996), GD, Springer, pp. 409–418. `doi:10.1007/BFb0021824`. 7, 9

[OKK13] OELKE D., KOKKINAKIS D., KEIM D. A.: Fingerprint Matrices: Uncovering the dynamics of social networks in prose literature. *Computer Graphics Forum 32*, 3pt4 (2013), 371–380. `doi:10.1111/cgf.12124`. 14, 15

[OM08] OGAWA M., MA K.-L.: StarGate: A unified, interactive visualization of software projects. In *Proceedings of the IEEE Pacific Visualization Symposium* (2008), PacificVis, IEEE, pp. 191–198. `doi:10.1109/PACIFICVIS.2008.4475476`. 14

[OM09] OGAWA M., MA K.-L.: code_swarm: A design study in organic software visualization. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1097–1104. `doi:10.1109/TVCG.2009.123`. 14

[PB08] POHL M., BIRKE P.: Interactive exploration of large dynamic networks. In *Proceedings of the 10th International Conference on Visual Information Systems* (2008), VISUAL, Springer-Verlag, pp. 56–67. `doi:10.1007/978-3-540-85891-1_8`. 7, 9

[PHG07] PURCHASE H. C., HOGGAN E., GÖRG C.: How important is the "mental map"? – an empirical investigation of a dynamic graph layout algorithm graph drawing. In *Graph Drawing*, GD. Springer, 2007, pp. 184–195. `doi:10.1007/978-3-540-70904-6_19`. 13

[PS08] PURCHASE H. C., SAMRA A.: Extremes are better: Investigating mental map preservation in dynamic graphs. In *Diagrammatic Representation and Inference*, Diagrams. Springer,

2008, pp. 60–73. `doi:10.1007/978-3-540-87730-1_9`. 13

[PT10] PUPYREV S., TIKHONOV A.: Analyzing conversations with dynamic graph visualization. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications* (2010), ISDA, IEEE, pp. 748–753. `doi:10.1109/ISDA.2010.5687175`. 14

[RD10] REY G. D., DIEHL S.: Controlling presentation speed, labels, and tooltips in interactive dynamic graphs. *Journal of Media Psychology: Theories, Methods, and Applications 22*, 4 (2010), 160–170. `doi:10.1027/1864-1105/a000021`. 13

[Rei10] REITZ F.: A framework for an ego-centered and time-aware visualization of relations in arbitrary data repositories. *arXiv preprint arXiv:1009.5183* (2010). URL: `http://arxiv.org/abs/1009.5183`. 7, 11, 14

[RM13] RUFIANGE S., MCGUFFIN M. J.: DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2556–2565. `doi:10.1109/TVCG.2013.149`. 7, 12, 13, 15

[RPD09] REITZ F., POHL M., DIEHL S.: Focused animation of dynamic compound digraphs. In *Proceedings of the 13th International Conference on Information Visualisation* (2009), IV, IEEE Computer Society, pp. 679–684. `doi:10.1109/iv.2009.24`. 7, 9

[RUK*10] ROHRSCHNEIDER M., ULLRICH A., KERREN A., STADLER P. F., SCHEUERMANN G.: Visual network analysis of dynamic metabolic pathways. In *Advances in Visual Computing*, ISVC. Springer, 2010, pp. 316–327. `doi:10.1007/978-3-642-17289-2_31`. 14

[SLN05] SARAIYA P., LEE P., NORTH C.: Visualization of graphs with associated timeseries data. In *Proceedings of the 2005 IEEE Symposium on Information Visualization* (2005), InfoVis, IEEE Computer Society, pp. 225–232. `doi:10.1109/INFVIS.2005.1532151`. 13

[SMM12] SALLABERRY A., MUELDER C., MA K.-L.: Clustering, visualizing, and navigating for large dynamic graphs. In *Graph Drawing* (2012), GD, pp. 487–498. `doi:10.1007/978-3-642-36763-2_43`. 7, 12

[SNF10] STAB C., NAZEMI K., FELLNER D.: SemaTime – timeline visualization of time-dependent relations and semantics. In *Advances in Visual Computing*, ISVC. Springer, 2010, pp. 514–523. `doi:10.1007/978-3-642-17277-9_53`. 14

[SP08] SAFFREY P., PURCHASE H.: The "mental map" versus "static aesthetic" compromise in dynamic graphs: a user study. In *Proceedings of the 9th Conference on Australasian User Interface* (2008), AUIC, Australian Computer Society, Inc., pp. 85–93. 13

[SQ07] SHANNON R., QUIGLEY A. J.: Considerations in dynamic graph drawing: A survey, 2007. URL: `http://rossshannon.com/publications/softcopies/rs2007-dynamic-graphs-survey.pdf`. 3

[STT81] SUGIYAMA K., TAGAWA S., TODA M.: Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics 11*, 2 (1981), 109–125. `doi:10.1109/tsmc.1981.4308636`. 9

[SWS10] STEIN K., WEGENER R., SCHLIEDER C.: Pixel-oriented visualization of change in social networks. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining* (2010), ASONAM, IEEE, pp. 233–240. `doi:10.1109/asonam.2010.18`. 7, 11

[SWW11] SHI L., WANG C., WEN Z.: Dynamic network visualization in 1.5D. In *Proceedings of the 2011 IEEE Pacific Visualization Symposium* (2011), PacificVis, pp. 179–186. `doi:10.1109/PACIFICVIS.2011.5742388`. 7, 11

[TK05] TOYODA M., KITSUREGAWA M.: A system for visualizing and analyzing the evolution of the web with a time series of graphs. In *Proceedings of the 16th ACM Conference on Hypertext and Hypermedia* (2005), HYPERTEXT, ACM, pp. 151–160. `doi:10.1145/1083356.1083387`. 14

[TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247–262. `doi:10.1006/ijhc.2002.1017`. 10

[Tri06] TRIER M.: Towards a social network intelligence tool for visual analysis of virtual communication networks. In *Virtuelle Organisationen und Neue Medien* (2006), TUDPress, pp. 331–342. 14

[Tuf06] TUFTE E. R.: *Beautiful Evidence*, 1st edition ed. Graphics Press, 2006. `doi:10.1177/0021943607306139`. 11

[VBSW13] VEHLOW C., BURCH M., SCHMAUDER H., WEISKOPF D.: Radial layered matrix visualization of dynamic graphs. In *Proceedings of the 17th International Conference Information Visualisation* (2013), IV, pp. 51–58. `doi:10.1109/IV.2013.6`. 7, 12

[vdEHBvW13] VAN DEN ELZEN S., HOLTEN D., BLAAS J., VAN WIJK J.: Dynamic network visualization with extended massive sequence views. *IEEE Transactions on Visualization and Computer Graphics* (2013). `doi:10.1109/TVCG.2013.263`. 7, 11

[VHK*13] VEHLOW C., HASENAUER J., KRAMER A., RAUE A., HUG S., TIMMER J., RADDE N., THEIS F. J., WEISKOPF D.: iVUN: interactive visualization of uncertain biochemical reaction networks. *BMC Bioinformatics 14*, Suppl 19 (2013), S2. `doi:10.1186/1471-2105-14-S19-S2`. 14, 15

[vLDBF13] VON LANDESBERGER T., DIEL S., BREMM S., FELLNER D. W.: Visual analysis of contagion in networks. *Information Visualization* (2013). `doi:10.1177/1473871613487087`. 14, 15

[vLKS*11] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J. J., FEKETE J.-D., FELLNER D. W.: Visual analysis of large graphs. *Computer Graphics Forum 30*, 6 (2011), 1719–1749. `doi:10.1111/j.1467-8659.2011.01898.x`. 1, 3, 15

[WZF11] WINDHAGER F., ZENK L., FEDERICO P.: Visual enterprise network analytics – visualizing organizational change. *Procedia – Social and Behavioral Sciences 22* (2011), 59–68. `doi:10.1016/j.sbspro.2011.07.056`. 3

[YAPM08] YANG X., ASUR S., PARTHASARATHY S., MEHTA S.: A visual-analytic toolkit for dynamic interaction graphs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2008), KDD, ACM, pp. 1016–1024. `doi:10.1145/1401890.1402011`. 14

[YEL10] YI J. S., ELMQVIST N., LEE S.: TimeMatrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction 26*, 11-12 (2010), 1031–1051. `doi:10.1080/10447318.2010.516722`. 7, 11, 15

[ZKS11] ZAMAN L., KALRA A., STUERZLINGER W.: The effect of animation, dual view, difference layers, and relative re-layout in hierarchical diagram differencing. In *Proceedings of Graphics Interface 2011* (2011), GI, Canadian Human-Computer Communications Society, pp. 183–190. 13

Biography

Fabian Beck is a postdoctoral researcher at VISUS, University of Stuttgart, Germany. In 2013, he obtained his Dr. rer. nat. (PhD) degree in computer science from University of Trier. In his dissertation, he investigated multi-dimensional coupling graph structures of software systems. In general, most of his research is focused on methods for visualizing and comparing large and dynamic graphs and hierarchies, often in the context of software systems and their evolution.

Michael Burch is a postdoctoral researcher at VISUS, University of Stuttgart, Germany. He received his Dr. rer. nat. (PhD) degree in computer science from the University of Trier in 2010. His main research interests are in information visualization, in particular, visualizing dynamic directed and weighted graphs with an additional hierarchical organization of the nodes in a static diagram. Other areas of expertise are the large-scale visualization of hierarchical data, eye-tracking evaluation of information visualization techniques, and the visualization of eye-tracking data.

Stephan Diehl is a full professor at University of Trier, Germany. He received his Dr. rer. nat. (PhD) degree as a DFG scholar at Saarland University, Germany, in 1996. Since 2007, he is a member of the scientific directorate of Schloss Dagstuhl – Leibniz-Center for Informatics. His main research areas are software engineering and information visualization, and in particular the intersection of both areas, namely software visualization. A considerable part of his visualization research can be briefly characterized as applying (visual) data mining techniques to analyze software evolution and developing new ways to visualize the change of structured information over time, for instance, as dynamic graphs.

Daniel Weiskopf is a full professor at VISUS, University of Stuttgart, Germany. He received his Dr. rer. nat. (PhD) degree in physics from the University of Tübingen, Germany, and the Habilitation degree in computer science at the University of Stuttgart, Germany. His research interests include all areas of visualization, visual analytics, GPU methods, perception-oriented computer graphics, and special and general relativity.