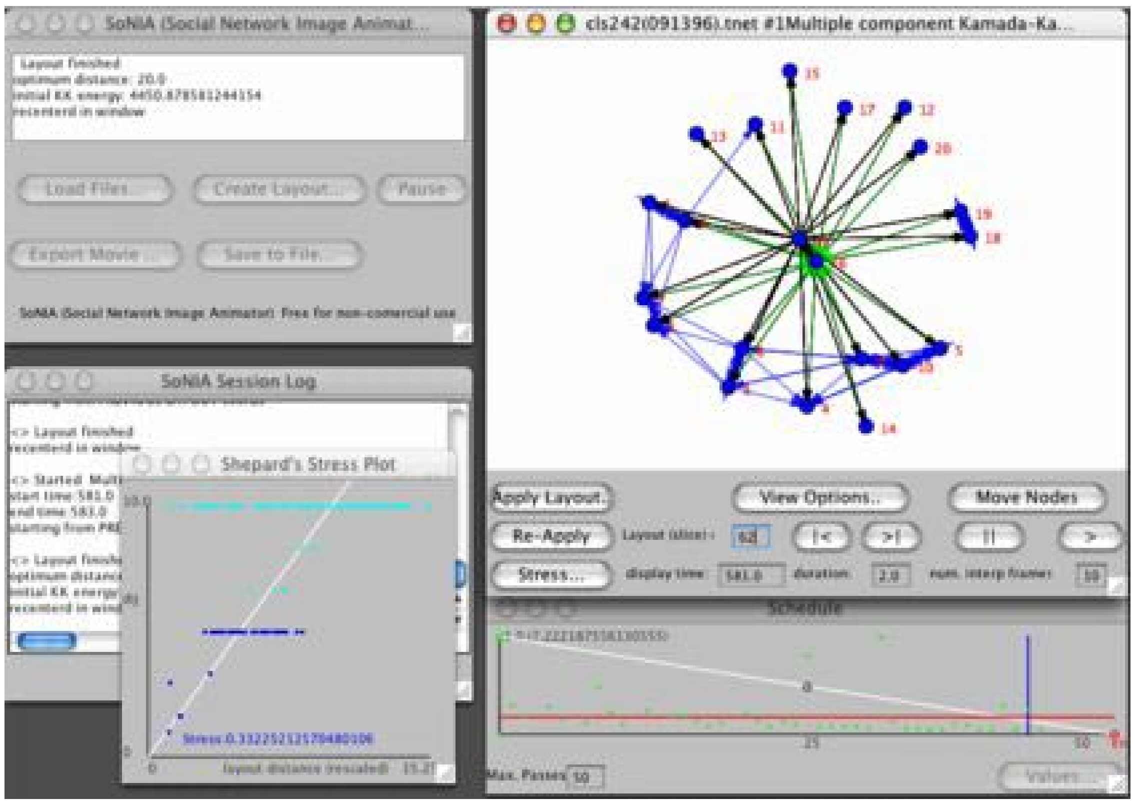
Interaction, Time, and Motion: Animating Social Networks with SoNIA

Skye Bender-deMoll and Dan McFarland



Abstract:
We consider some issues which arise with the aggregation of continuous-time relational data ("streaming" interactions) to form network series. We demonstrate SoNIA (Social Network Image Animator) as a tool for constructing animations of dynamic networks (continuous or discrete), browsing attribute-rich network data, and as a platform for testing and comparing layouts and techniques. We discuss strengths and weakness of existing layout algorithms and suggestions for adapting them to sequential layout tasks.

<http://sonia.stanford.edu>

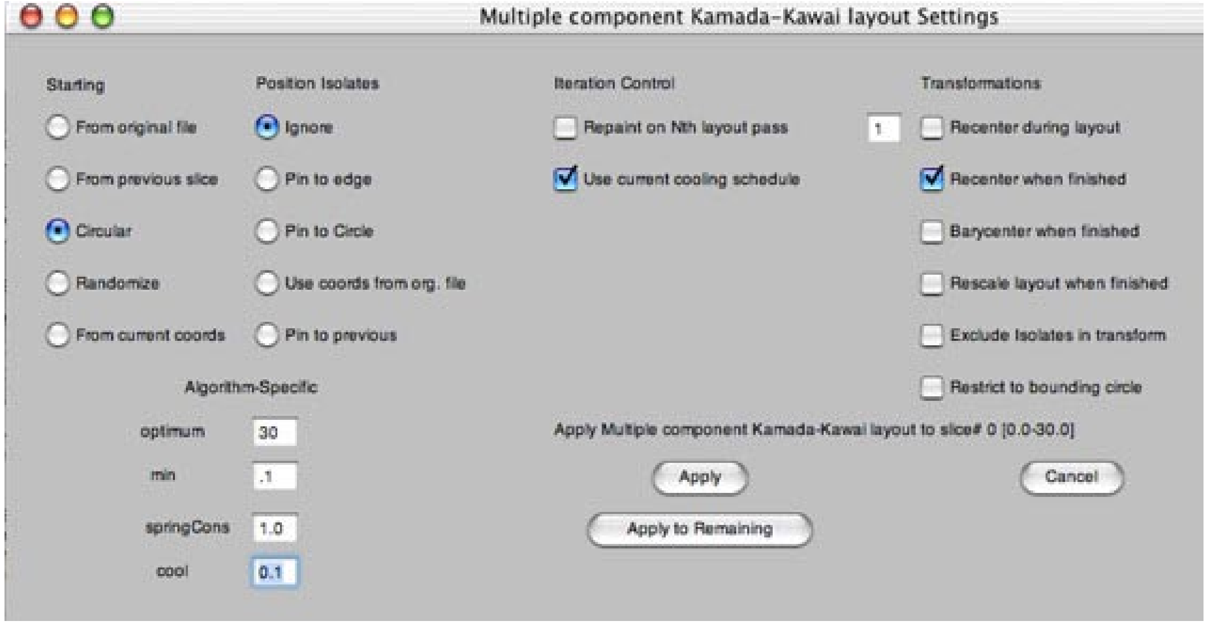
SoNIA was originally developed by Dan McFarland and Skye Bender-deMoll (**skyebend@stanford.edu**) , generously supported by a Research Incentive Award provided by Stanford University's Office of Technology and Licensing (grant# 2-CDZ-108).

Introduction:

SoNIA is a Java-based prototype package for visualizing dynamic or longitudinal "network" data.

In theory, it will work on all platforms which can run Java

By dynamic network data, we mean that in addition to information about the relations (ties) between various entities (actors, nodes) there is also information about when these relations occur, or at least the relative order in which they occur.



Goals for SoNIA:

- explicitly deal with time in visualization
- aid user in constructing accurate meaningful layouts
- platform for testing algorithms , and exploring parameters
- browser for attribute rich time based network data
- focus on doing visualization well, not a general SNA package
- work flexibly with data in various time formats

Demo movie: McFarland classroom network

Network movie representation of the **“streaming” interaction data** in one of the classrooms observed by Dan McFarland in 1996. (Which cite dan?)

Each interaction observed is represented by a directed arc.

Statements that were directed to everyone appear as a "star" of links.

Direct interactions were weighted as 1 (thick lines) and indirect or overheard interaction were weighted as 1/nth.

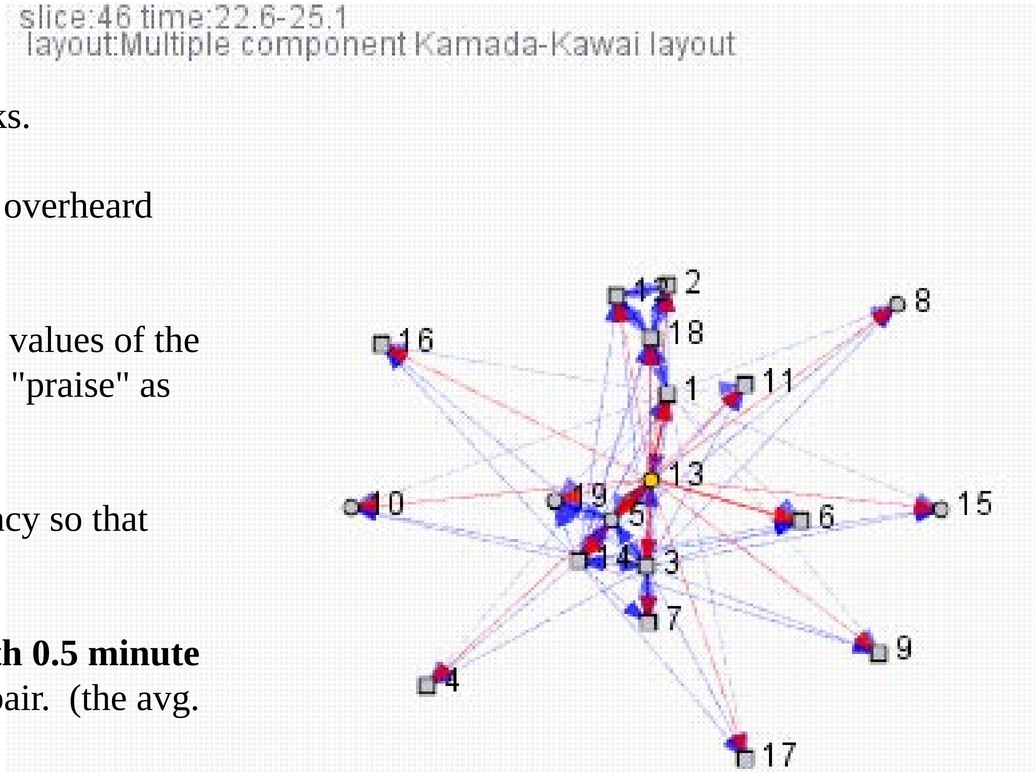
Coding of "social", "task" and "sanction" to control the RGB color values of the ties. "task" interactions appear black, "social" interactions as blue, "praise" as green and "sanction" as red, "social-sanction" as purple, etc.

To help show this visually, ties are rendered with partial transparency so that underlying ties show through to some degree.

This network is shown with **slices of 2.5 minutes in duration, with 0.5 minute slice delta**, so there may generally multiple ties between a given pair. (the avg. of their weights is used)

Multi-Component Kamada-Kawai algorithm (1) with SoNIA’s modified cooling procedure, Started from circle, subsequent layouts chained, default cooling parameters

Additional movies are at <http://sonia.stanford.edu/exampleMovies/>



MOVIE 6. Class re683, 12/10/96 – English 10, Composition: Maintenance, Seatwork, and Rebellion “...teacher is overrun by her students and finds it nearly impossible to get them to adhere to the routine of seatwork at some minimum level. “

(1) Kamada, K. and Kawai, S. (1989) “An Algorithm for Drawing General Undirected Graphs.” *Information Processing Letters* 31, 7-15

Static vs. Dynamic Networks

Very few data sets are truly “dynamic”, most have only a few waves (1)

Even in multi-wave studies, information on duration and ordering of events in unavailable because data is compressed into network matricies

Large amounts of time-network data now available from automated collection and simulation work.

Are the higher level patterns nothing more than the sum of the micro-events?

How much time is “enough” to be able to describe change in a network?

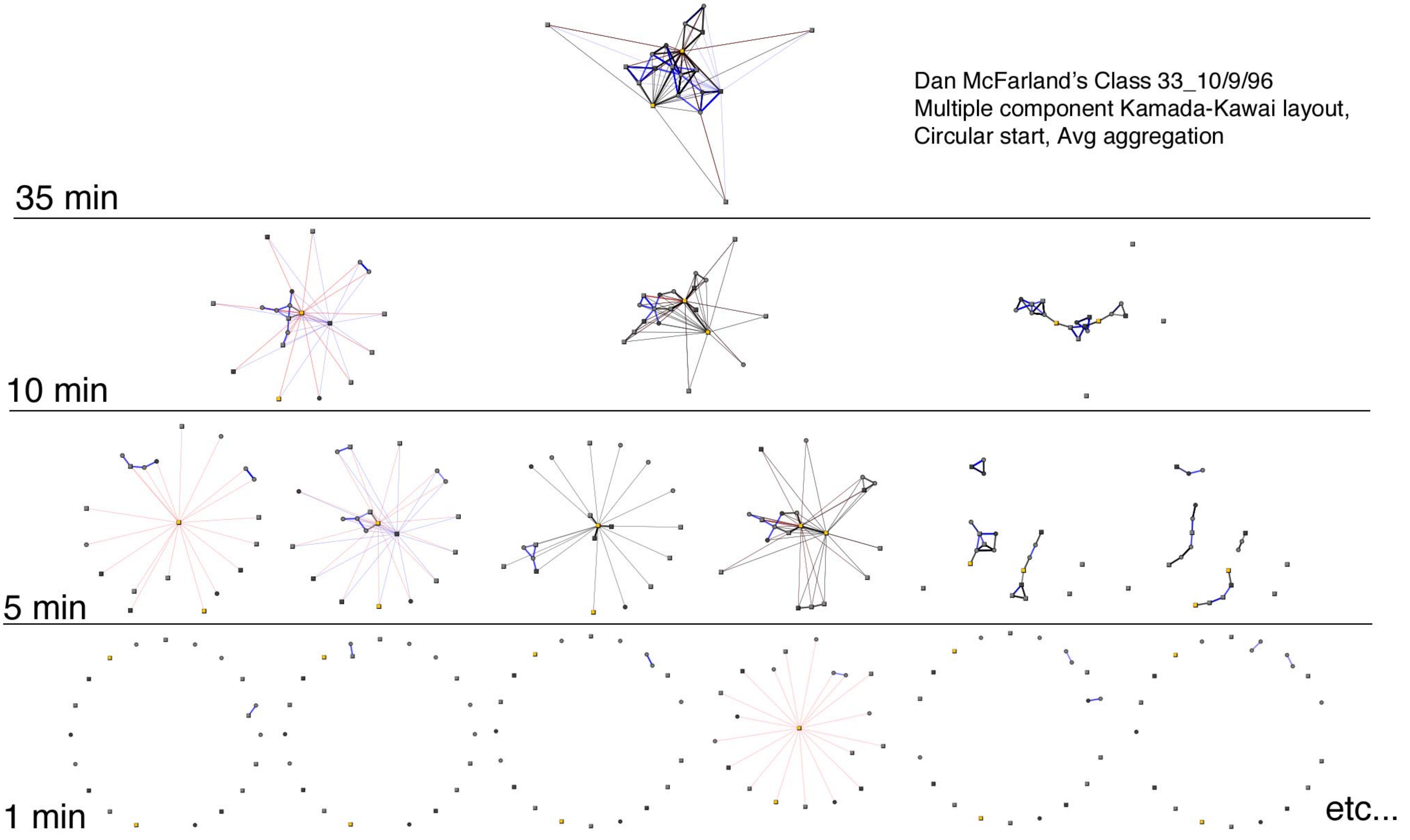
At what level are dyad- and triad-level effects (activity, reciprocity, etc) relevant?

At what scale do network-level effects (cliquing, transitivity) appear?

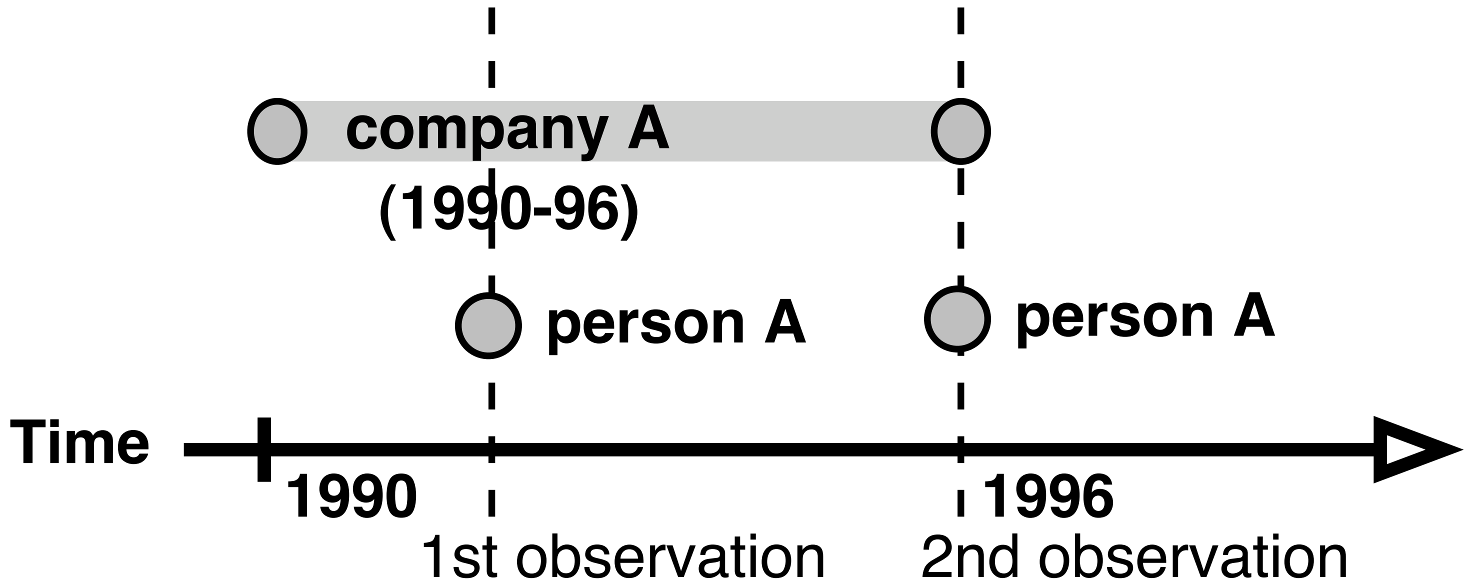
Static	compressing all the data into one network = lots of range and detail
Slow	breaking into multiple networks = less detail, but ability to look at change
Fast	events / streaming approach = loose network entirely, but can look at ordering

(1) Doreian, P., and Stokman, F. (1997) “The Dynamics and Evolution of Social Networks” *Evolution of Social Networks* ed. Patrick Doreian and Frans Stokman. Gordon and Breach. 1-17.

Static vs. Dynamic Networks



Unpacking the Matrix



Consider the entities (nodes, individuals) as a set of events.

Every event has a real-valued time coordinate indicating when it occurs.

If the event is not instantaneous, it also has an ending coordinate to indicate its duration.

Events can describe attribute or position changes.

Matrix data can be integrated by importing it as a large group of simultaneous events

Matrices can be generated flexibly from event data using a slicing (binning) procedure

Slice & dice

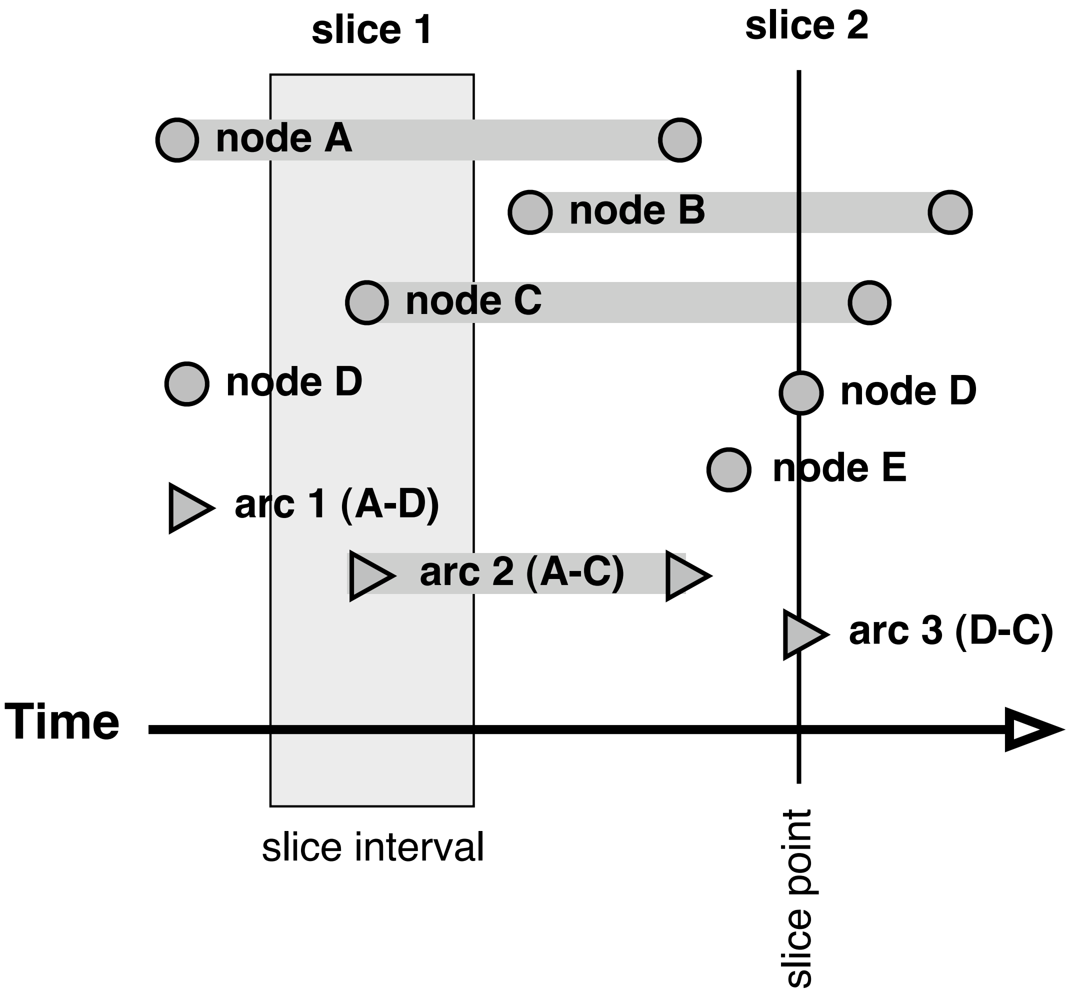
To capture the changes in a network over time, a series of slices can be used. The appropriate number of slices and their duration depends on the network of interest.

Consider the sampling period of the data, the rate of sampling, rate of change of the variable, and rate of change of network of interest

Thick vs. thin slice
Thick = interval, includes nodes within or intersected, used with instantaneous data or "pre-binned" data "**show me all the loans among firms that took place *during* 1994**."

Thin = instant, includes what it intersects, use with duration data. "**show me the network *at* time t.**"

Because multiple arcs can occur between a pair during an interval, an aggregation technique must be used to get the actual arc weights

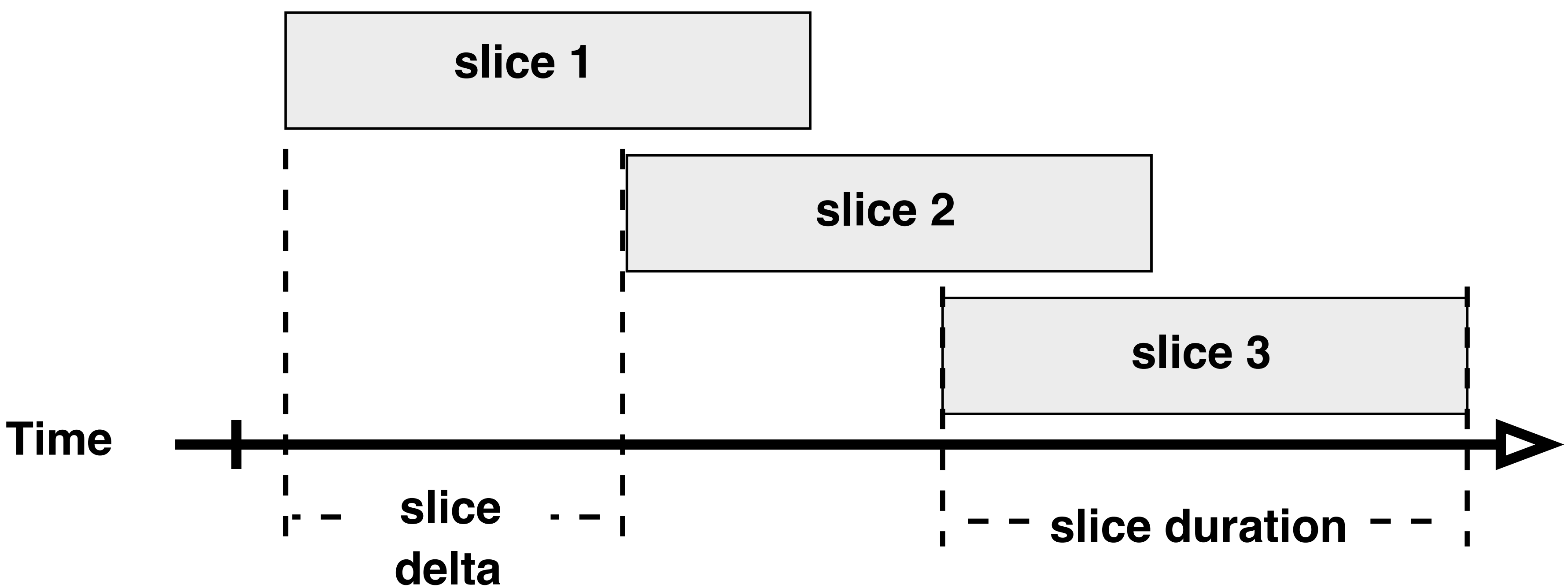


Slice & dice

Useful to bin networks using a "**sliding window**", incrementally moving a slice of a certain width along the timeline.

Functions like a **rolling average**, adding a few new ties to the front of the slice and dropping the old ones as the start and end times are incremented by a small delta.

Letting the bins overlap can be thought of as a crude way of allowing events to "decay", or act as if they have a **limited duration** or window of effect on the network.



Browsing Network Time Data Demo

If the underlying data are "streaming" or **un-binned event data**, very **short duration render slices** can be used to browse or tease-apart fine scale interactions, inspecting them almost tie-by-tie.

In SNA there are obvious practical limitations to the time resolution which can be achieved with survey instruments, such as memory effects of respondents, test-retest reliability, etc. (1), some of these problems are not present in observational and automated data collection techniques, or output from network simulations.

(1) Ferligoj, A., Hlebec, V. (1999) "Evaluation of Social Network Measurement Instruments" *Social Networks* 21 (1999) 111-130
Marsden, P.V. (1990) "Network Data And Measurement" *Annual Review of Sociology* 16 435-463.

Continuous vs. Discrete time

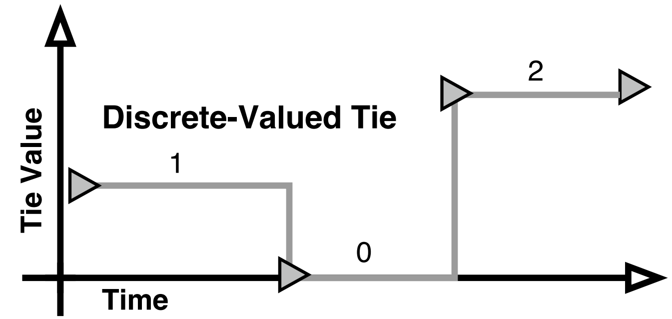
Discrete time methodology
Sampling the network variables at **regular intervals**, recording the value of each tie at every sample point, a matrix for each wave.

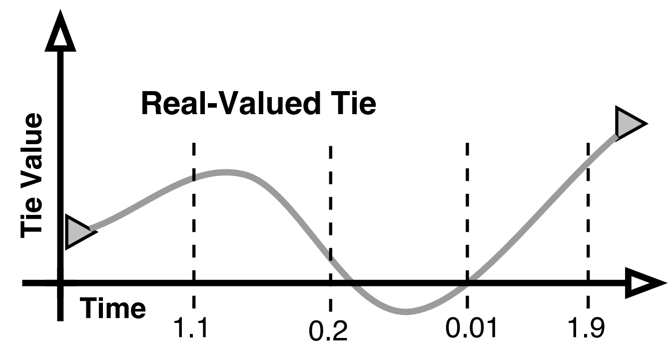
Continuous time methodology
Record the **real-valued time coordinate** for each state change or event. The entire network is not observed at once, rather a “stream” of - usually asynchronous - tie changes are recorded, along with the time coordinate of the transition

Both time and tie values can be discrete or real-valued. Confusing because “discrete events” implies “continuous time”

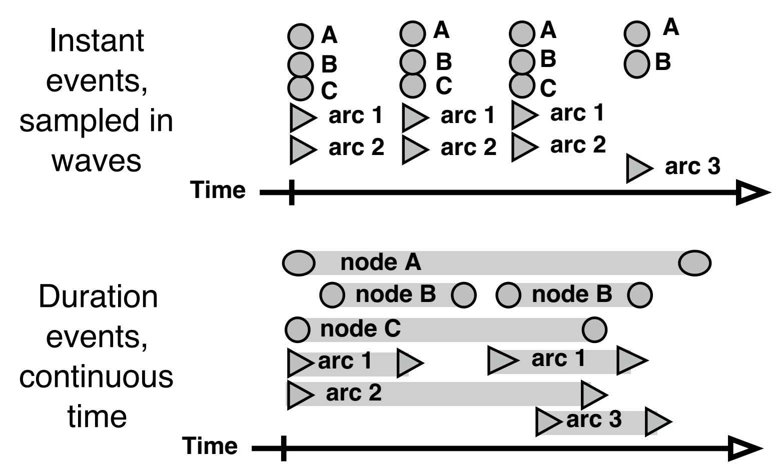
The contrast between the different approaches becomes apparent if we try to assign time coordinates to the data. The **discrete** approach breaks time down into a series of **chunks or blocks** within which all finer graduations are irrelevant. If we are using **discrete** intervals, “**1 to 2**” means **two units of time**, where in the **continuous** approach, **1 to 2** could either mean 1.0 to 2.0 (**one unit of time**) or 1.0 to 2.99.. (very close to two units of time)

In SoNIA, both kinds of time can be used, but it is crucial to be clear what is meant in the underlying data.





Similar network with two strategies of time:



Social Cartography

Ideally, network visualization might serve a role similar to geographical cartography, if its strengths and limitations are well understood and techniques widely used, they become transparent, allowing us to concentrate on the relationships revealed rather than on the tools for presenting them.

(2)

Maps serve to communicate information about the relationships and distances between geographical entities by **discarding information in order to convey a larger abstraction**. In order to make this compression possible, choices must be made about **which geometric relationships to preserve, and which to discard or distort.**

Network visualization has the power to mislead and distort as well as to inform - it is well established that the positioning of nodes on a layout has a large impact on the intuitive conclusions the viewers will draw (1,3)

(1) McGrath, C. Blythe, J., Krackhardt, D. (1997) "Seeing Groups in Graph Layouts" http://www.andrew.cmu.edu/user/cm3t/groups.html
(2) Piran images copied without permission from http://tis.telekom.si/karta/map.exe and http://dragonja.nib.si/indexA.html
(3) Blythe, J., C. McGrath, and D. Krackhardt. 1995. "The Effect of Graph Layout on Inference from Social Network Data." *Symposium on Graph Drawing GD 95*. Passau.

Social Cartography

Embedding dimension (2)

When network data is conceptualized as a social space, its "natural" dimensionality may be far higher than the three or four used to depict physical space, so a **zero-stress 2D layout is usually impossible**. (This can create stopping time problems for algorithms)

Stress and Shepherd's plots

A modified version of Kruskal's "Stress" function (1) compares the matrix of "desired distances" between nodes to the actual screen distances produced by the layout to asses the distortion. Only appropriate for some layout algorithms.

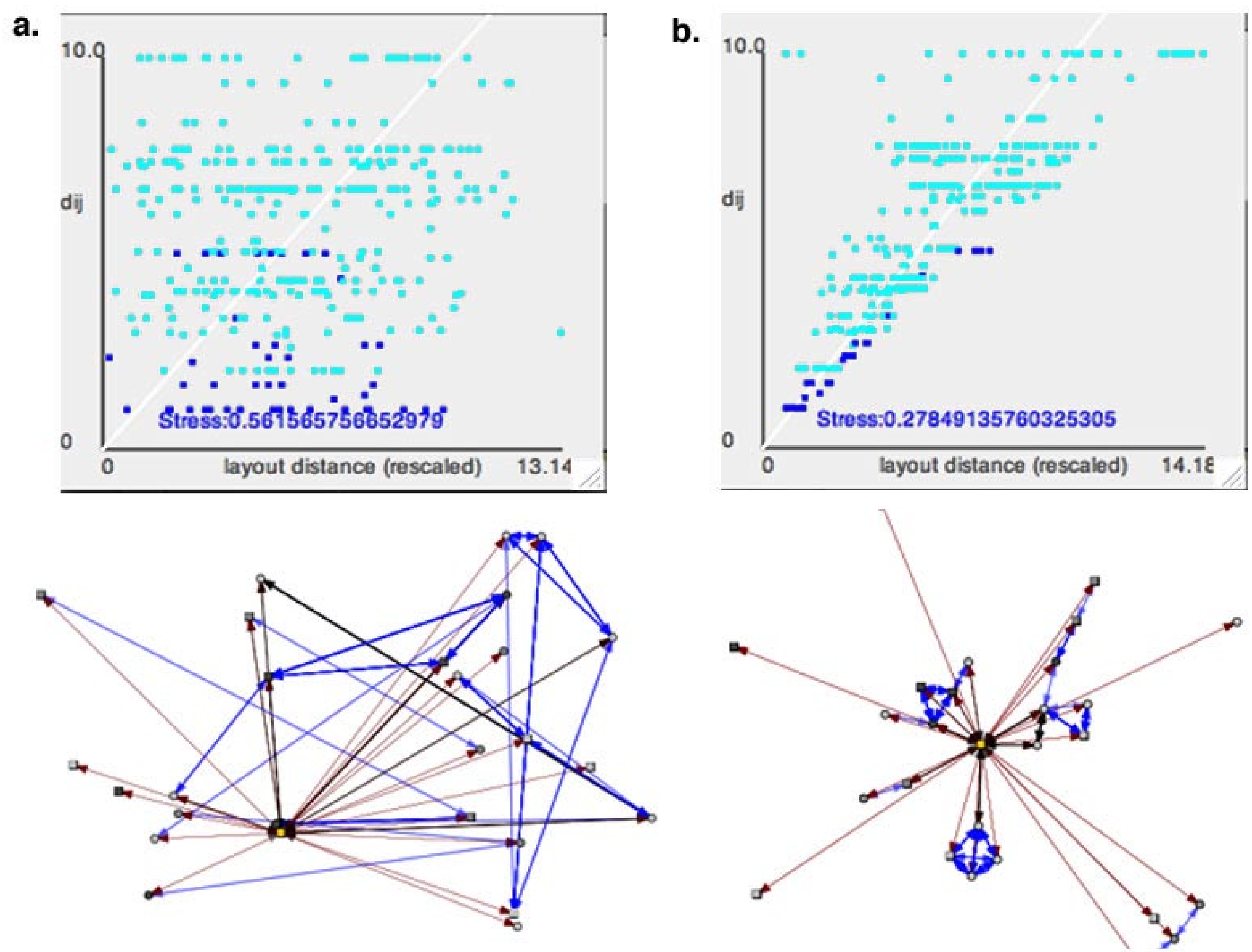


Figure a shows a layout with nearly random node positioning and its Shepherd's stress plot. Figure b shows the results of running the KK algorithm. In b, the points are much more clustered around the line of equality (white diagonal) and the stress level is lower, indicating a closer correspondence of geodesic and geometric distance.

(1) Kruskal, J., Wish, M. (1978) *Multidimensional Scaling*. Sage Publications. Beverly Hills. CA.
(2) Freeman, L. (1983) "Spheres, Cubes and Boxes: Graph Dimensionality and Network Structure" *Social Networks* 5:2 139-156

Networks and social space

Almost all data require transformation before they can be mapped to visualizeable social space

We can’t expect visualization techniques to give stable, consistent, or useful results unless the social space we are trying to visualize is itself stable and consistent.

Each domain will have individual criteria of what constitutes a “good” layout.

Some useful general criteria to consider before attempting to create a visualization of a network (that is not to say we always follow them well!):

1. What is the underlying set of relations we are really interested in looking at and how can they be best expressed?

2. What is the functional relationship between the collected data and the relations of interest?

3. What time-scale are the patterns of interest likely to be visible at?

4. How might node and arc attributes relate to the pattern of network structure, and how can they best be translated into display variables in order to hi-light and explore these relations? (1)

5. What set of transformations do we need to apply to get from the data to a consistent social space?

(1) Tufte, E., (1983) *The Visual Display of Quantitative Information*. Graphics Press

From Matrix to Metric

Aggregation technique. - to create matrix from multiple and multiplex ties in slice Sum, Avg, Count (rate). Aggregation raises an interesting question where events are considered simultaneous, (and therefore statistically independent to some degree) when in fact they may be ordered and directly dependent.

Similarity vs. dissimilarity (1)
Similarity (**larger values** mean more similar and **closer placement** on the layout) must be reversed to dissimilarity (**larger values** mean more **distant placement**). To be consistent across slices, SoNIA uses the transformation equation:

(1 / tieValue) * maxValue

where maxValue is the maximum tie value in any of the slice networks (after aggregation, excluding infinity).

Symetrization - in a metric space, distances must be symmetric
Normally use max value.

Geodesic distance and All-Pairs-Shortest Path
KK uses geo-dist, FR (1) uses raw distances.

Components and isolates
Disconnected components create an infinite distance problem (or repulsion in FR)
Run algorithm on each component, options for ignoring or treating isolates separately, “post processing”

(1) Scott, J. (2000) *Social Network Analysis*. Sage, Newbury Park CA
(2) Fruchterman, T., Reingold, E. (1991) “Graph Drawing by Force-directed Placement” *Software-Practice and Experience*, Vol 21(11), 1129-1164.

Rendering & Animation

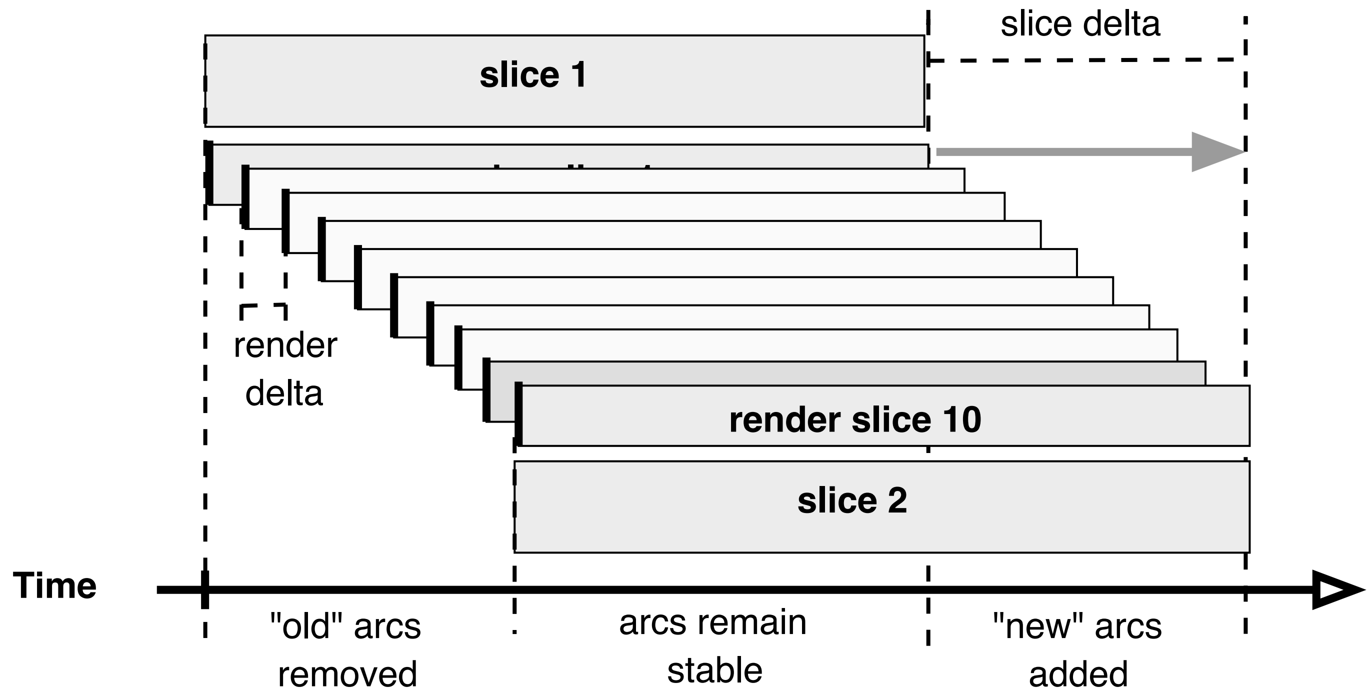
Use animation to preserve the "mental map" (1)

Cosine interpolation (slow-in, slow-out, natural motion) (2)

Maintain consistent time-scale
The duration of the animation between two slice networks should be proportional to the time between the networks in the data. Can be difficult with complex graphics, which is why exporting a movie can be useful.

Coordinate interpolation vs. network interpolation
1) Create a layout for each network, and then create a series of "tweening" images of the network in which the nodes' ***layout coordinates*** **are interpolated** to create the impression of a gradual transition from the first to the second.

2) Create a series of intermediary "interpolated" networks, in which the values of the ***arc weights*** **are interpolated** and the layout algorithm reapplied to generate the intermediate images. (This may require a very stable algorithm)



SoNIA currently uses #1, generating a series of "render slices" which slide along the timeline from starting to ending slice, adding and dropping events as they pass through

(1) Branke, J. (2001) "Dynamic Graph Drawing" in M. Kaufmann and D. Wagner (Eds.): *Drawing Graphs*, LNCS 2025, 228-246. Springer-Verlag
(2) Yee, K., Fisher, D., Dhamija, R., Hearst, M. (2001) "Animated Exploration of Graphs with Radial Layout" *IEEE Symposium on Information Visualization* 43–50.

Techniques for time

1. Small slice delta = small Hamming Distance
Using overlapping slices (slice delta << slice duration) with “un-binned” streaming data makes it possible to generate networks which are very similar, with only a **few arc changes**, which greatly simplifies the layout task, and makes the process easier to follow.

2. Pick good starting coordinates for the first slice
Can help prevent components overlap, establish a “baseline” for viewer.

3. Chaining layout starting coordinates - starting each one from the coordinates of the previous.
If there is little change in the network between the two slices, the algorithm may have little work to do to find a new optima, and it is likely to be fairly close spatially to the previous one. However, there is an increased chance of remaining on local optima.

4. Apply transformations consistently
Re-scale is generally a bad idea, adjust layout parameters instead.

5. Reduce or eliminate meaningless movement.
The human eye is well adapted to detecting and classifying motion (1). This makes animation a powerful tool because, if done well, it allows the natural tendencies of our perceptual hardware to aid in the intuitive analysis of the data, detecting areas of interest, relative rates, outliers, etc. Unnecessary movement is distracting.

6. Use animations as a complement for statistical and other means of conveying the data.

7. Design algorithms which use information from previous layouts for constraint
rubber bands, etc.

(1) Gibson, J. (1986) *The Ecological Approach to Visual Perception.* Lawrence Erlbaum.

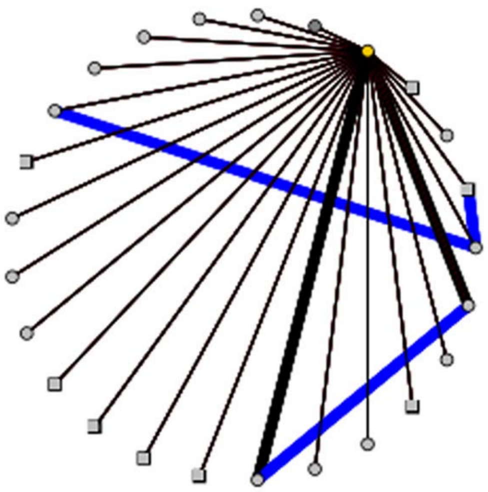
Graphics & Making movies

Java's graphics framework makes it possible to do some nice things with images:

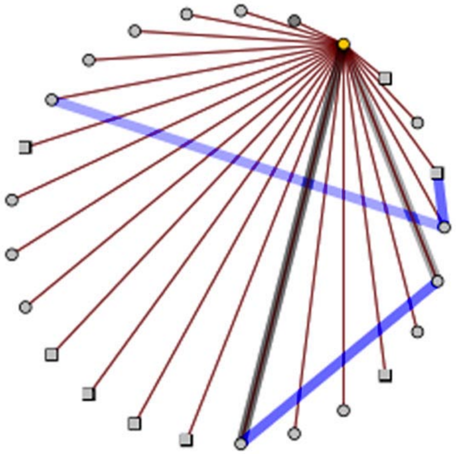
Transparency (so arcs can show through)

Compositing (using a ghost image of a previous layout to asses layout changes)

Anti-Aliasing (for smooth graphics)



Without Transparency



With Transparency (0.2)

Recording QuickTime Movies
Apple's QuickTime QTJava library makes it possible to export movie files of animations.

Movie files play at a fixed rate, so deals with problems of graphics slowing down playback on complex graphs.

Files are based on a compressed bitmap format, so they are very large.

Simple to put up on the web, QuickTime player is available for Macs and PCs and is integrated w/ browser.

Would be good to have a vector animation format like Macromedia's Flash (.swf) file format.

Need to add ability to write Animated GIFs, PostScript, JPEGs, etc.

Good idea to save log file with movie so layout can be reconstructed later.

Data formats

.SON (SoNIA format)

Arc list format, node definitions, then arcs,

Tab-delimited with **column headings** order insensitive (except Ids)

Multiple records for same node or arc are possible (to **describe attribute changes**)

Error messages are specific to the line and entry, and describe the acceptable values.

.NET "[ ]" time format

Limited implementation for compatibility with Pajek (1)

Attribute mapping & SQL

Should map variables like PajekConverter (2)

Front end for SQL database?...

Other network attribute formats (VNA?)

```
//McFarland Classroom 15_10/7/96
NodeId  Label    StartTime EndTime  NodeSize   NodeShape  ColorName  BorderWidth B
1       129473   0.0       42.0     5.0        ellipse    lightGray  1.5         b
2       129047   0.0       42.0     5.0        rect       gray       1.5         b
3       132996   0.0       42.0     5.0        ellipse    lightGray  1.5         b
4       145242   0.0       42.0     5.0        ellipse    gray       1.5         b
5       127535   0.0       42.0     5.0        ellipse    lightGray  1.5         b
6       127319   0.0       42.0     5.0        rect       lightGray  1.5         b
7       129801   0.0       42.0     5.0        ellipse    darkGray   1.5         b
8       104456   0.0       42.0     5.0        ellipse    lightGray  1.5         b
FromId ToId  StartTime EndTime        ArcWeight         ArcWidth     ColorName
24      1     0.135     0.135  0.2     1.6      black
24      2     0.135     0.135  0.2     1.6      black
24      4     0.135     0.135  0.2     1.6      black
24      3     0.135     0.135  0.2     1.6      black
24      6     0.135     0.135  0.2     1.6      black
24      5     0.135     0.135  0.2     1.6      black
24      7     0.135     0.135  0.2     1.6      black
24      9     0.135     0.135  0.2     1.6      black
24      8     0.135     0.135  0.2     1.6      black
24      11    0.135     0.135  0.2     1.6      black
24      10    0.135     0.135  0.2     1.6      black
24      12    0.135     0.135  0.2     1.6      black
16      25    3.514     3.514  0.2     1.6      black
16      26    3.514     3.514  0.2     1.6      black
```

(1) Batagelj, V., Mrvar, A. (1998, 2004) *Pajek - Program for Large Network Analysis* http://vlado.fmf.uni-lj.si/pub/networks/pajek/

(2) Bender-deMoll, S. PajekConverter - a utility for converting tab-delineated text files into a format readable by the network analysis and visualization software Pajek. http://student.bennington.edu/~skyebend/pajekConvert.htm

Conclusions

1) SoNIA is a prototype which will require additional development, testing, and documentation before it will serve as a stable cross domain research tool, but is fairly unique in its focus on time-network data and has considerable potential.

2) Although a great deal more work is needed to create reliable layouts and animations, SoNIA demonstrates that the **animation approach is feasible**.

3) Adapting static layout procedures to dynamic tasks is not a trivial problem.

4) This kind of visualization may make possible **“Petri Dish” SNA**, allowing us to get a much better intuitive grasp of network processes - which will hopefully translate into better models.

5) We need to think very carefully about time when recording network data, and work towards more **fine-scaled approaches to datasets**.

6) Additional work on integrating algorithms and expertise generated in other domains (energy minimization, dimensional projection) since the “standard” algorithms were written is needed, and may bring much better results.

7) **Flexible binning of event data** makes it possible to explore the inflection points between micro and macro level network processes.

Future of the program

To date, SoNIA has mostly be used to visualize streams of interaction data from small work groups or classrooms, needs testing on lager more complex data sets.

We’ve done research on algorithms we’d like to work with, (GEM (1), Magnetic Force Layout (2), Non-Metric MDS, High-Dimensional Embedding and Projection (3), etc.) interface improvements, needed features (export files, beef up movies, SWF, multiple layout stress assessment, etc.)

Debate about making available under GNU GPL (OpenSource) license, or a “free for academic use / license for commercial use” arrangement. Depends if other people are interested in helping with development.

Commented source and API docs will always be available to researchers who ask. (as soon as they are done!)

For more information, visit **http://sonia.stanford.edu** or contact **skyebend@stanford.edu**

(1) Frick ,A., Ludwig, A., Mehldau, H. (1994) “A Fast Adaptive Layout Algorithm for Undirected Graphs”, *Proceedings of the DIMACS International Workshop on Graph Drawing*, 388-403
(2) Sugiyama, K. and Misue, K. (1995) “A Simple and Unified Method for Drawing Graphs: Magnetic-Spring Algorithm” *Lecture Notes in Computer Science* 894, Springer Verlag, 364-375 .
(3) Harel, D. and Koren, Y. (2002) “Graph Drawing by High-Dimensional Embedding” *Graph Drawing, 10th International Symposium, Revised Papers, Lecture Notes in Computer Science 2528.* Springer