Design Alternatives for Storyline Visualization of Multivariate Time Series

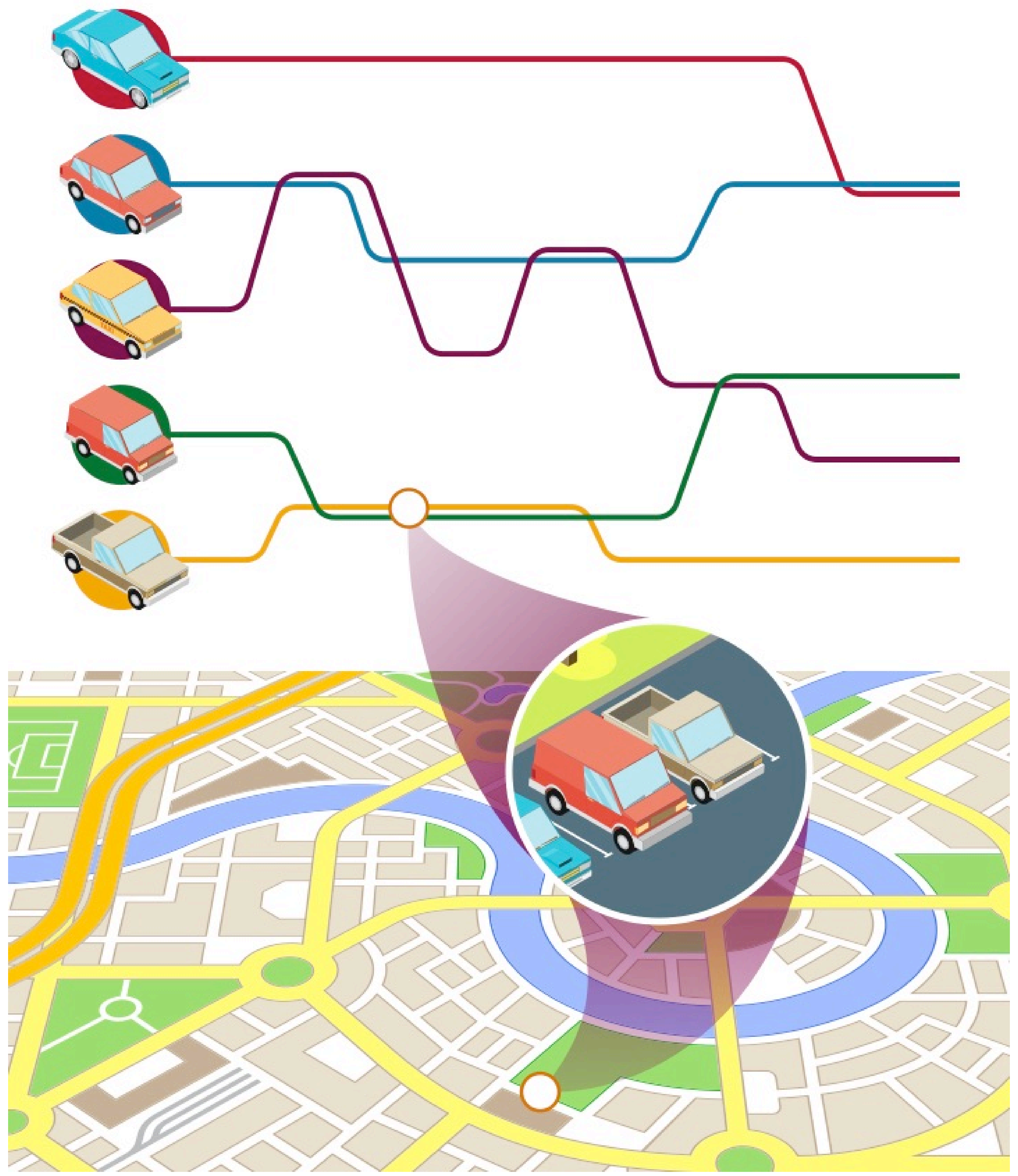
Dustin Arendt (Visual Analytics, National Security Directorate)

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle *Since 1965*

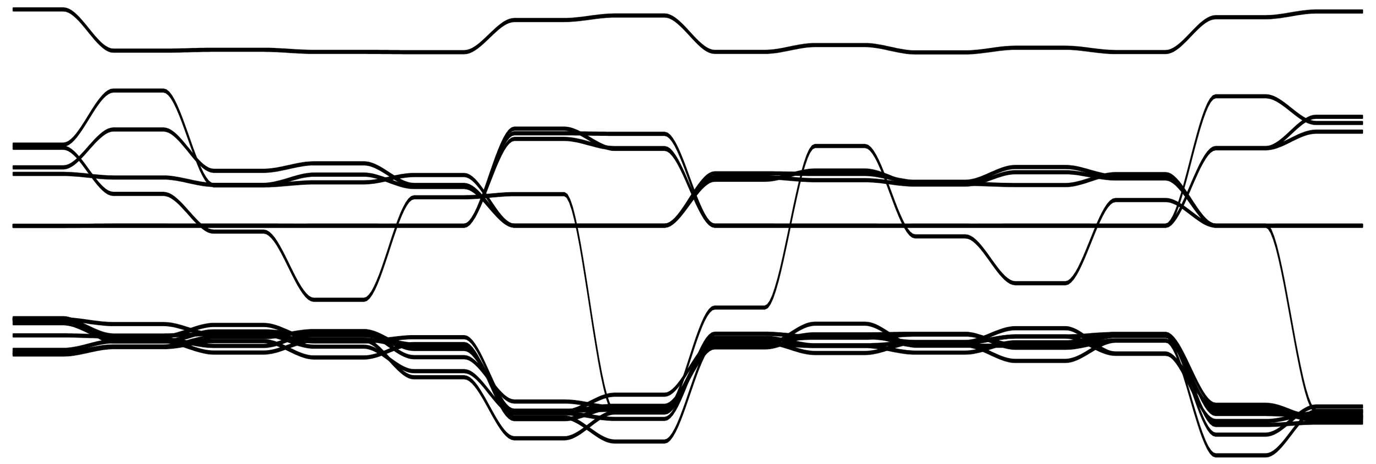
SEDP
Scientist & Engineer Development Program

Storyline Visualization

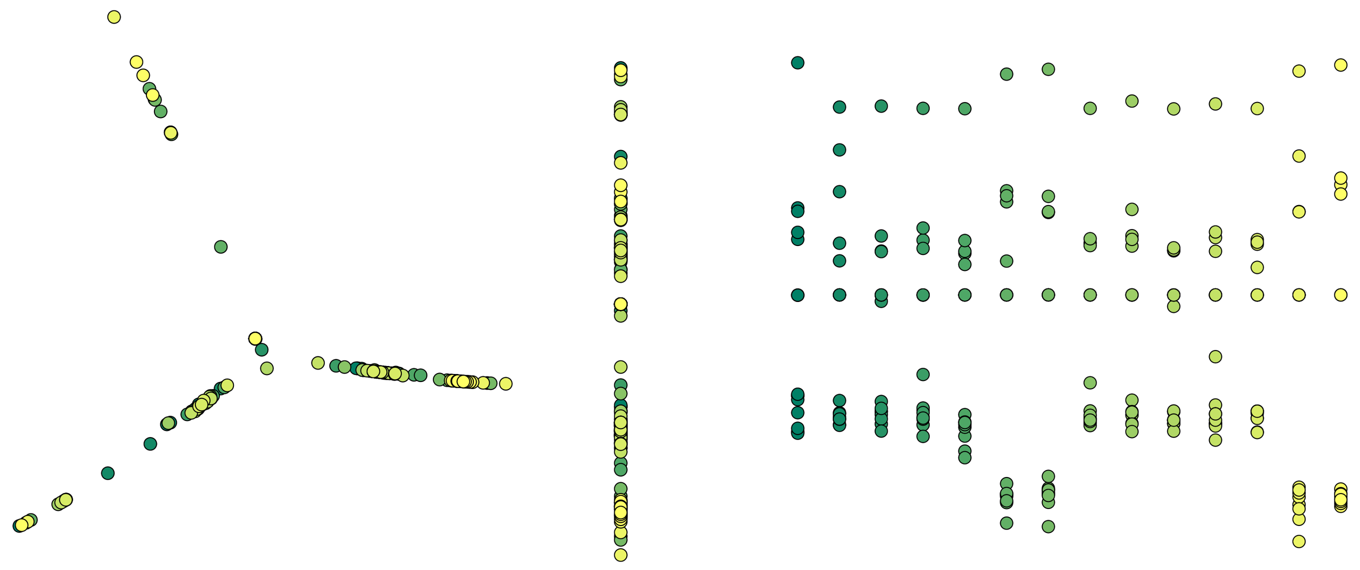
Patterns, relationships, and anomalies in time-varying data are revealed through exploratory visual analytics using storylines. Each entity is represented as a line, and time is encoded on the horizontal axis. When entities are similar at a given time point, their storylines come together, otherwise they are drawn apart. This poster explores some alternative algorithms for determining the y-coordinate of the storylines.



Alternative 1: Projected



Manifold learning techniques are used to project high dimensional data into lower dimensional spaces. Points that are close in the original space should also be close in the lower dimensional space. Often data is projected into 2 dimensions so it can be visualized (a). The storyline layout can be found by projecting the time series data into a 1-D space (b) and then plotting that against time (c).

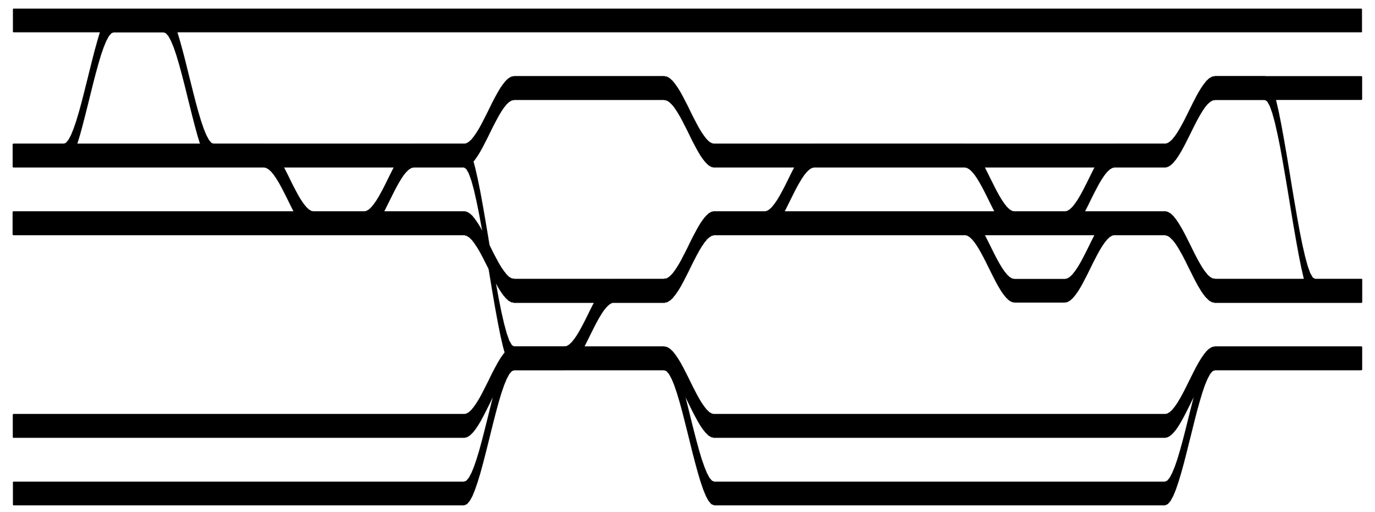


(a) 2-D *(b) 1-D* *(c) 1-D vs. time*

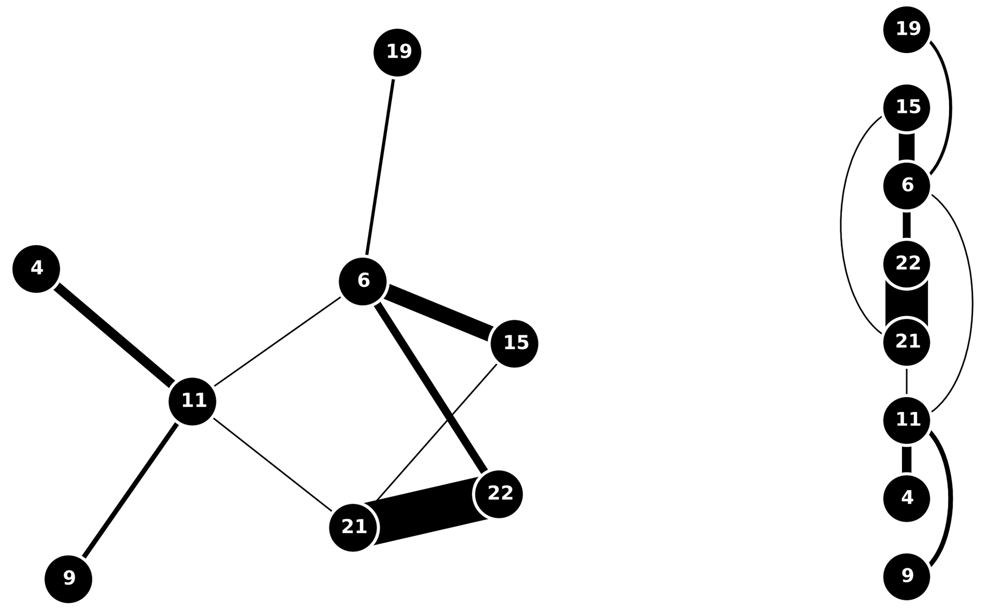
Easy to implement
Distance between lines may be meaningful

Over plotting (moderate)
Potentially too much noise

Alternative 2: Layered



Alternatively, if storylines are grouped using a clustering algorithm, we could uniquely assign y-coordinates to clusters. Doing so randomly would create unnecessary line crossings. We should assign layers in a way that reveals frequent transitions between groups. To do this, we compute the group flow graph (a) independent of time, and then find the spectral ordering of the nodes in this graph (b). This causes strongly connected nodes to be closer together.

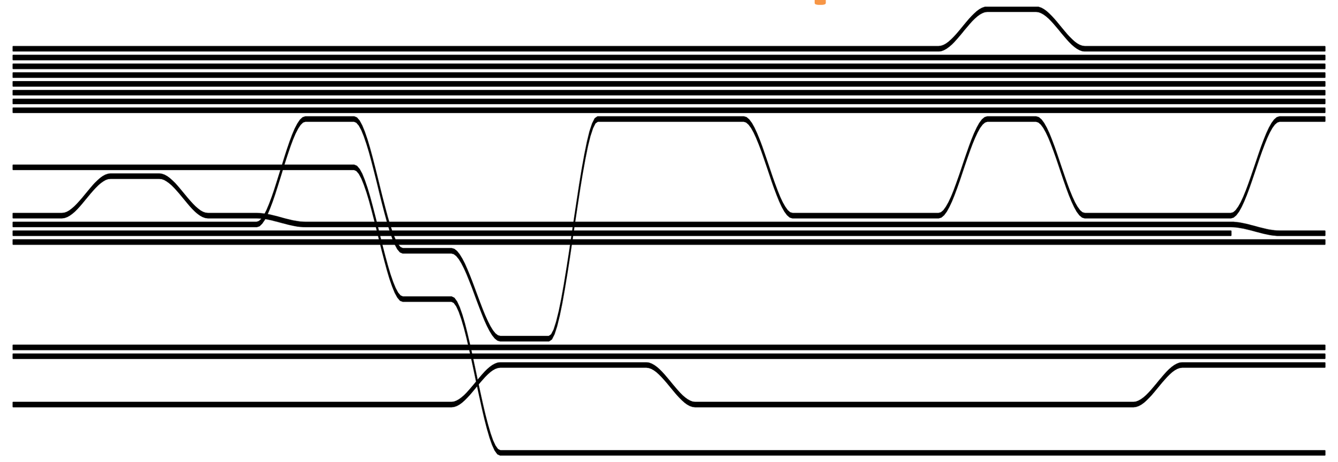


(a) Group flow *(b) Ordered flow*

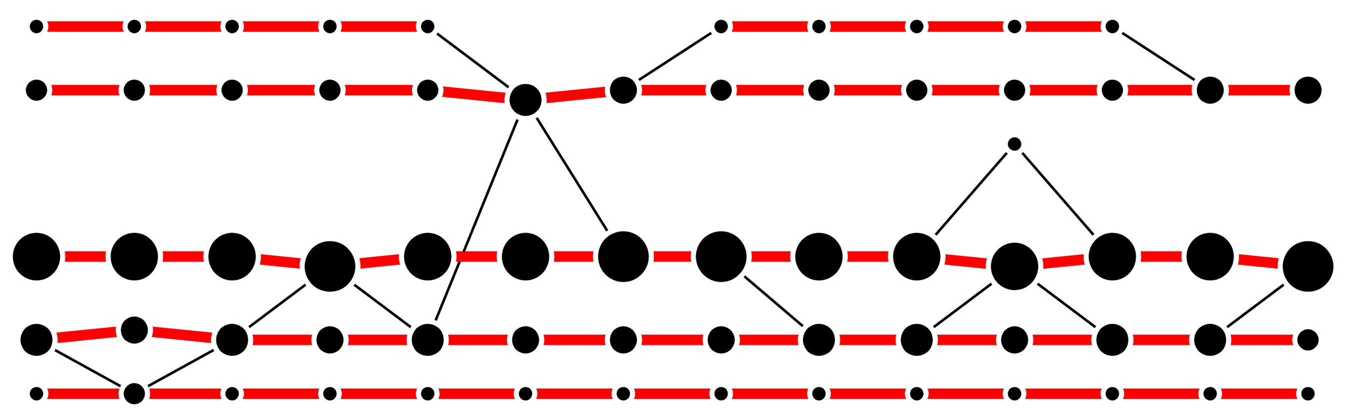
Easy to implement
Position of line (and deviation) is meaningful/consistent

Over plotting (severe)

Alternative 3: Optimized



Instead of letting it fall to chance, we can explicitly arrange the visualization to reduce line crossings and wiggles that would otherwise make the visualization difficult to read. We use a multi-stage optimization algorithm to accomplish this. In the first stage, the order of the groups is optimized, then, respecting this order, edges are aligned (shown as red lines below). Solving a linear program based on the previous constraints determines the y-coordinate of the storylines.



(a) Group flow graph

Minimal over plotting
Aesthetically pleasing

Difficult to implement
Line deviation might not be meaningful, causing confusion