Understanding Feature Evolution Over Time For Large-scale Time-varying Datasets

Wathsala Widanagamaachchi* Valerio Pascucci†

SCI Institute, University of Utah

ABSTRACT

Understanding temporal evolution of features has long been a problem of interest for a variety of applications in science and technology. *Tracking graphs* which show the evolution of features across time as a collection of tracks are the method of choice for representing such data. However, due to the ever increasing sizes of datasets, constructing and visualizing them in a comprehensible manner and performing interactive changes is challenging. Here, we present a research proposal which enables users to explore and understand these time-verying data sets, regardless of the underlying data type. As such, we present a novel visualization and analysis environment which enables interactive exploration of dynamically constructed tracking graphs.

1 PROBLEM

One of the most common analysis tasks is the need to understand the evolution of time-varying features. Often, there exists some notion of a feature-of-interest at each moment in time, e.g. burning cells in combustion data, twitter topics in social media data, and these features evolve over time. Exploring and analyzing the behaviors of these features with respect to changes in parameters, such as thresholds, and in time are of interest. However, this poses a significant challenge as it involves coupling the analysis both within and across timesteps. A temporal analysis multiplies the amount of data that must be considered simultaneously, making it challenging to present them in a comprehensive manner.

Abstract *tracking graphs* that indicate the evolution of features as a collection of tracks that split/merge over time are often used to represent the complex spatio-temporal relationships across such features. Creating a tracking graph requires two independent components: the ability to define a feature-of-interest; and a way to correlate them across time. For both aspects, there exists a wide range of solutions, such as clustering [4] or topological analysis [2] to define features, and spatial overlap- or predictor-based for feature tracking [14, 15, 12, 16]. Nevertheless, for the terabyte-scale datasets common today, constructing these tracking graphs often results in hours or days of file I/O time alone, making dynamic modification of feature parameters or correlation criteria infeasible. Furthermore, creating the corresponding optimal graph layouts which minimize the edge intersections may take hours to compute preventing any interactive changes. Finally, for all but the smallest datasets, such graphs, even assuming an optimal layout, quickly become incomprehensibly large and complex for users to understand. See Figure 1. These aforementioned reasons severely limit the ability to explore the relationships between feature selection parameters and the temporal evolution of features.

*e-mail: wathsy@sci.utah.edu

†e-mail: pascucci@sci.utah.edu

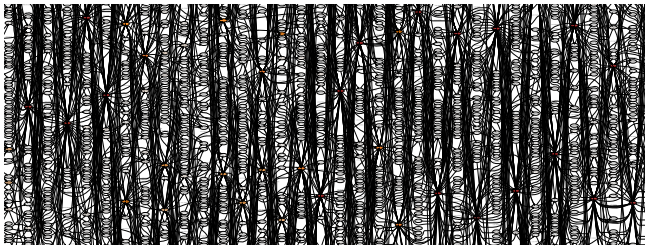

Figure 1: A zoomed in view of a tracking graph for a device scale combustion simulation dataset [1, 2] visualized and optimized using dot [6]. The displayed graph only contains 33 of the 324 timesteps (819 nodes), yet its still nearly incomprehensible due to the complex interactions.

2 DISSERTATION STATEMENT

In consideration of the above discussion, we propose that it is possible to dynamically define, extract and simplify tracking graphs for large-scale time-varying datasets. We further propose that it is possible to design an interactive framework for addressing the general task of understanding feature evolution over time without focusing on any specific application domain. As such, this proposed work leads to a new framework that couples hierarchical feature definitions with progressive graph layout algorithms to provide an interactive exploration of dynamically constructed tracking graphs. The utility and generality of our approach is demonstrated using several large-scale scientific and non-scientific data sets.

3 METHODOLOGY & PLAN OF RESEARCH

Defining & extracting dynamic tracking graphs. We intend to make use of flexible data representations, which are significantly smaller than the original datasets and yet retain enough information to perform the desired temporal analysis. As such, one data representations for storing the features within timesteps and the other for storing the correspondences are to be used. For the first, a feature hierarchy which encodes all possible features in a timestep for a wide range of parameter settings is used for storing the features within a timestep. Along with the hierarchy, various feature-based attributes like first order statistical moments and/or shape characteristics are also computed and stored. Secondly, a new compact and efficient meta-graph structure is introduced which, similar to the feature hierarchy for features, stores not one particular tracking graph but the entire family of tracking graphs for all possible feature parameters. Together, these data structures allow interactive extraction of tracking graphs for a particular set of parameters and correlations criteria.

Graph layout & visualization. As many datasets involve thousands of features for hundreds of timesteps, the sheer number of features existing within a tracking graph makes the graph drawing challenging. Nevertheless, we observe that one rarely needs to look at all features across all timesteps simultaneously. Accordingly, we propose to process the tracking graphs with respect to a focus timestep and a window of interest. This limits our focus of

interest to a certain sub-region of the global tracking graph. We also intend to develop progressive graph layout algorithms for computing the optimal layouts to reduce edge intersections. Here, we progressively layout the tracking graphs both forward and backward in time and the user is able interactively change timesteps, expand and contract the window of interest and thus explore the entire graph.

Reducing visual complexity of graphs. For many datasets of practical interest, even within a window of interest and with optimized layouts, the large number of features and their complex relationships can make the resulting tracking graphs nearly unmanageably large and difficult to comprehend. As a result, it is not only impossible to follow a given track through time but also difficult to identify the salient feature tracks within the graph. Therefore, we intend to provide several techniques to further filter and simplify these graphs. Specifically, we try to reduce nodes and edges in a graph so that its underlying patterns can be easily identifiable. For example, tracking graphs often contain spatially small features which are not necessarily of interest and many spurious merge and split events also exist that are distracting rather than being informative. Using two different approaches: filtering and feature selecting, we intend on allowing users to interactively sub-select the graph in both space and time. Furthermore, we also propose to stabilize the feature evolution over time and produce more temporally cohesive graphs by locally adapting the feature thresholds.

Framework design. For understanding time-varying features, apart from the methods for storing features and their correlation details, it is also essential to maintain an interactive visualization and analysis environment. We propose a visualization environment containing of three different views. The first two presenting general conceptual views of the time dependent feature hierarchies, the third presenting a more specialized view for feature embedding. Combined, these modules allow users to explore features, their evolution, and how different scales affect their behavior. Furthermore, the entire framework is to be implemented in a progressive fashion in which all parameter changes, graph manipulations, and layouts are to be computed interactively in a streaming fashion.

Framework generality. This proposed framework has a major advantage compared to existing approaches, due to the generality of the design. Specifically, both methods used for defining and tracking features are to be selected and/or modified to accept general feature hierarchies. We also allow any of the standard clustering algorithms to define (hierarchical) features and any existing correlation criteria, such as volume overlap or aggregated attributes, to define correlations. Therefore, various large-scale scientific and non-scientific data sets can be explored within the proposed framework to understand their underlying trends. In particular, we intend to make use of combustion, ocean science, cosmology and plasma surface interaction datasets in the scientific domain and social media and healthcare datasets within the non-scientific domain.

Pattern identification of graphs. Large and complex tracking graphs often contain motifs which are repeated throughout the graph either because of the feature behavior or the parameter selection. Regardless of the cause, identifying and analyzing these patterns can reveal interesting details and underlying trends of the data. On the other hand, some of these frequently occurring motifs contain little information compared to the space they occupy in the graph. In such cases, identifying them can lead to the simplification of the graph. In certain situations, users would like to understand how much a certain feature's evolution differs from other feature tracks in the graph and identifying the patterns within and across feature tracks is likely to facilitate this process.

Multi-variate feature hierarchies. Up to this point, the proposed work is mainly focused on understanding feature evolution of time-varying data with respect to single-variate feature hierarchies. For each timestep of a given dataset, its feature hierarchy is constructed by grouping the features at different scales and this feature grouping is always constructed based on a single parameter, e.g temperature in burning cells, textual similarity in tweets. Little research has being done on constructing and analyzing feature hierarchies for two or more properties [3]. We intend to explore on this topic of constructing multi-variate feature hierarchies and extend the proposed framework to handle such hierarchies.

4 Progress to Date

Here, a description of the effort and progress on the proposed research plan is presented. The section is partitioned into content from various publications resulted from my involvement on the project.

1. *Interactive Exploration of Large-Scale Time-Varying Data using Dynamic Tracking Graphs, W. N. Widanagamaachchi, C. Christensen, P.-T. Bremer and V. Pascucci, Proceedings of IEEE symposium on Large-Scale Data Analysis and Visualization (LDAV), Seattle, USA, 2012.* [19]

In this paper, we first introduce our framework for interactively exploring feature evolution in massive time-dependent datasets. The framework is an interactive linked-view system which combines a tracking graph layout and a traditional 3D feature display, and is implemented using the ViSUS framework [9, 10]. We utilize topological merge trees for storing the feature hierarchy within timesteps and introduce a new meta-graph structure for encoding families of tracking graphs. Together, these data structures enable users to interactively define and extract local graphs.

We make use of progressive techniques to maintain the interactivity during the graph visualization. Consequently, the graph is always processed with respect to a user-defined focus timestep. Then, starting from this timestep, nodes and edges are iteratively added up to the user-defined window of interest. The graph layout is also computed as the graph is being created. However, as computing a optimal or near-optimal layout is expensive for larger graphs, we use two different strategies. The user is immediately presented with a fast initial graph layout which is replaced with a slower greedy layout (with less edge intersections) as soon as it is available. Here, we make use of the median heuristic for computing the greedy layout.

We also enable changing feature definitions on-the-fly and filter the graphs using arbitrary feature-based attributes while providing an interactive view of the resulting tracking graphs. Finally, we test and validate our framework with several large-scale scientific simulations from combustion science.

2. *Data-Parallel Halo Finding with Variable Linking Lengths, W. N. Widanagamaachchi, P.-T. Bremer, C. M. Sewell, L.-T. Lo, J. Ahrens and V. Pascucci., Proceedings of IEEE symposium on Large-Scale Data Analysis and Visualization (LDAV), Paris, France, 2014.* [18]

Here, we present a novel algorithm for constructing feature hierarchies for cosmological data. A "halo" [8] is an over-densed region of dark matter particles and represents one of the common features-of-interest within these datasets. One of the two most common definitions of a halo is friends-of-friends (FOF) clustering [5] where all particles that are reachable through links shorter than a predefined distance (the linking length) are considered to be one halo.

We make use of the PISTON library [7] and bring out a data-parallel, friends-of-friends (FOF) halo finding algorithm for creating a halo hierarchy for a range of linking lengths. Furthermore, we couple our algorithm to an interactive analysis environment to study halos at different linking lengths and track their evolution over time. Compared to all existing FOF-based halo finding algorithms which require re-computation of halos whenever the linking length or halo size parameters are changed, our algorithm has an significant advantage. The feature hierarchy we compute encodes

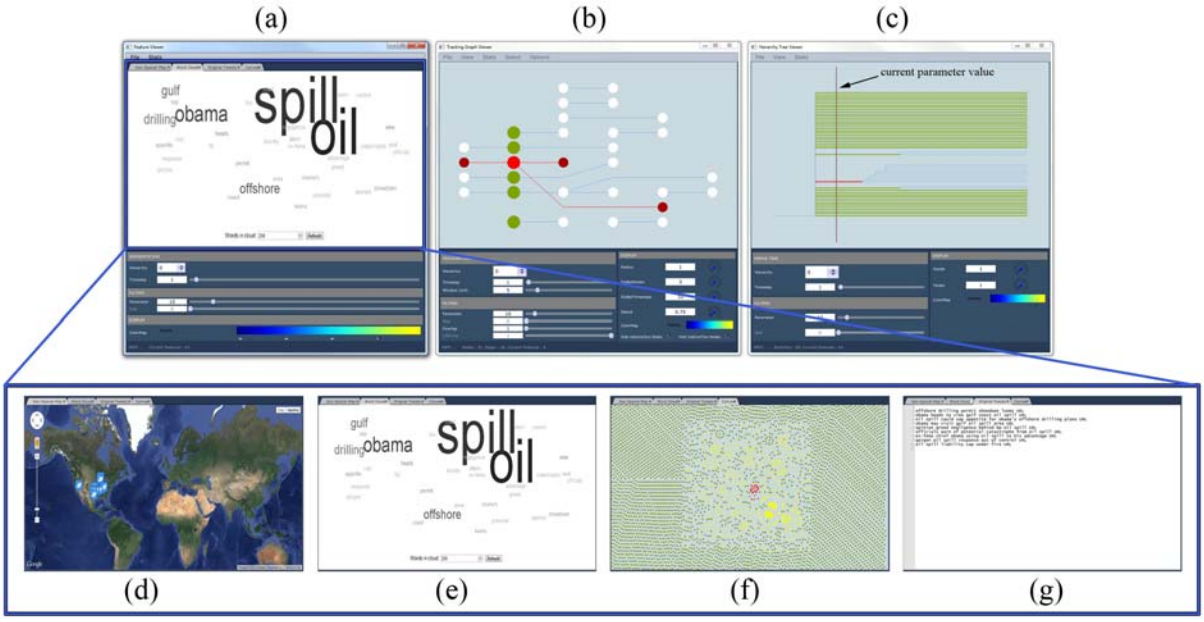

Figure 2: Our framework contains a view for (a) feature embedding, (b) feature tracking, and (c) feature hierarchy. The feature embedding view consists of several sub-components: (d) geospatial, (e) word cloud, (f) geometric embedding, and (g) textual views. Here, a social media data set is used and the selected feature and its track are indicated in 'red'.

all possible halos for a wide range of linking lengths. Therefore, it has the capability to quickly and efficiently compute halos for a wide range of linking length and halo size parameters without any re-computation.

Finally, the research conducted within this paper is used to extend the applicability of our aforementioned framework for exploring feature evolution over time. Specifically, the feature hierarchies computed from our halo finding algorithm are used along with a newly computed meta-graph (where correspondences are computed using a volume overlap-based correlation criteria) for interactively exploring the halo evolution over time.

3. *Visualization and Analysis of Large-Scale Atomistic Simulations of Plasma-Surface Interactions, W. N. Widanagamaachchi, K. Hammond, L.-T. Lo, B. Wirth, F. Samsel, C. M. Sewell, J. Ahrens and V. Pascucci., Proceedings of EuroVis - Short Papers, Cagliari, Italy, 2015. (To appear.)* [20]

Due to the interest in the origin of fuzz-like, microscopic damage to tungsten and other metal surfaces by helium, plasma-surface interaction simulations have recently been the focus of significant research. In this paper, a simulation-visualization pipeline for creating a visualization and analysis environment for atomistic simulations of plasma-surface interactions is presented. LAMMPS Molecular Dynamics Simulator [11] and the Visualization Toolkit (VTK) [13] has been used for creating this pipeline.

Simulations show that helium spontaneously aggregates to form clusters and eventually bubbles, pushing out tungsten surface defects, i.e. voids/cavities. The visualization phase of our pipeline identifies and visualizes the boundaries between helium-filled regions, tungsten-filled regions, and voids, using an algorithm encoded through library calls to VTK. The analysis phase identities several atom statistics and helium bubble evolution details through calls to LAMMPS.

Again, the research conducted within this paper is used to extend the applicability of the framework for exploring feature evolution over time. Here, we visualize the computed helium bubble evolution details within our framework and allow scientists to understand the evolution details of helium bubbles by exploring its parameter range.

4. *Understanding Feature Evolution over Time using Dynamic Tracking Graphs, W. N. Widanagamaachchi, P.-T. Bremer and V. Pascucci, Proceedings of IEEE transactions on Visualization and Computer Graphics, Chicago, Illinois, USA, 2015. (Submitted.)* [17]

In this paper, we mainly focus on improving and extending our framework for exploring feature evolution over time. Although the framework presented in [19] is able to provide interactive exploration of tracking graphs, there still remained a number of open challenges for applying it to the next generation of even larger datasets. One is the graph drawing which remains restricted by the limited number of nodes and edges a human can reasonably understand. Therefore, simplifying the graphs and selecting subgraphs allows users to easily perceive the graphs and their underlying trends.

Here, we introduce a new three-pass layout algorithm to optimize and simplify tracking graphs by exploiting a fuzzy parameter selection. Within a user-specified parameter range, this algorithm locally adapts the feature definition parameters and produces more temporally cohesive graphs. Specifically, using feature stability over time as a criterion, we optimize the graph in a greedy fashion and reduces the total number of non-valence two nodes.

Additionally, using two different approaches, filtering and feature selecting, we allow users to extract local graphs in both space or time. Filtering significantly reduces the complexity of the graph by lowering both node and edge count without losing any pertinent information. Here, we allow filtering based on any feature-based or edge-based attributes and the feature track length. For feature selecting, we enable the capability to spatially sub-select features and their tracks by screening according to their respective bounding boxes. However, as it is often useful to concentrate on a particular feature and its evolution rather than a particular region in space, we also provide the additional ability to select a certain feature in

the focus timestep and then use the graph connectivity to extract all related tracks both forward and backward in time.

Within this paper, we have generalized our framework by modifying the tracking graph creation approach of [19] to accept general hierarchies rather than only for spatial segmentation. The visualization environment is also modified to contain three different views: feature evolution, feature hierarchy and a feature embeddings view. See Figure 2. For the feature embeddings view, several visualization techniques (geometric, geospatial, word cloud and textual visualizations) are combined to present a specialized view of features. Finally, several new datasets from both scientific and non-scientific domains, e.g. ocean science, social media, are used for demonstrating the generality and versatility of our approach.

5 Timeline

The table 1 shows the intended timeline for this aforementioned proposal. The research work that has been accomplished, the remaining work and the expected time to complete them, is given in individual sections.

Defining & extracting dynamic tracking graphs			
✓	01.	Feature hierarchy construction	
✓	02.	Meta-graph construction	
✓	03.	Tracking graph generation	
✓	04.	Extending both data structures to accept general hierarchies	
Graph layout & visualization			
✓	05.	Initial layout computation	
✓	06.	Greedy layout computation	
✓	07.	Interactive visualization	
Reducing visual complexity of graphs			
✓	08.	Tracking graph sub-selection	
✓		**a.** Filtering	
✓		**b.** Feature selection	
✓	09.	Cohesive graph creation based on adaptive thresholding	
Framework design			
✓	10.	Data visualization	
✓	11.	Data exploration	
✓	12.	Implementation	
Framework generality			*2 months*
✓	13.	Scientific datasets	
✓		**a.** Combustion	
✓		**b.** Ocean science	
✓		**c.** Cosmology	
✓		**d.** Plasma-surface interaction	
	14.	Non-scientific datasets	
✓		**a.** Social media	
		b. Healthcare	
Pattern identification within graphs			*3 months*
	15.	Motif identification	
	16.	Graph simplification	
Multi-variate feature hierarchies			*4 months*
	17.	Multi-variate feature hierarchy construction	
	18.	Adding support to the current framework	

Table 1: Timeline.

6 Input Expected from Panelists

Listed below are couple of inputs I would appreciate to gain from the Vis Doctoral Colloquium panelists with regard to the aforementioned research proposal.

- Feasibility of the current research plan
- Modifications & improvements to proposed research work
- Applicability of methods to other areas of research
- Relevance to existing methods & research problems
- Potential extensions & applications

References

[1] B. Bedat and R. K. Cheng. Experimental study of premixed flames in intense isotropic turbulence. *Combust. Flame*, 100:485–494, 1995.

[2] P.-T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, Mar. 2010.

[3] H. Carr and D. Duke. Joint contour nets. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1100–1113, 2014.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[5] M. Davis, G. Efstathiou, C. S. Frenk, and S. D. White. The evolution of large-scale structure in a universe dominated by cold dark matter. *Astrophysical Journal*, 292:371–394, 1985.

[6] E. Koutsofios and S. North. Drawing graphs with dot. Technical Report 910904-59113-08TM, AT&T Bell Laboratories, Murray Hill, NJ, 1991.

[7] L.-t. Lo, C. Sewell, and J. Ahrens. Piston: A portable cross-platform framework for data-parallel visualization operators. In *Eurographics Symposium on Parallel Graphics and Visualization*, 2012.

[8] J. Navarro and S. D. White. The structure of cold dark matter halos. In *International Astronomical Union Symposium*, volume 171, pages 255–258. Kluwer Academic Publishers Group, 1996.

[9] V. Pascucci, P.-T. Bremer, A. Gyulassy, G. Scorzelli, C. Christensen, B. Summa, and S. Kumar. *Advances in Parallel Computing, Cloud Computing and Big Data*, volume 23, chapter Scalable Visualization and Interactive Analysis Using Massive Data Streams, pages 212–230. IOS Press, 2013.

[10] V. Pascucci, G. Scorzelli, B. Summa, P.-T. Bremer, A. Gyulassy, C. Christensen, S. Philip, and S. Kumar. *High Performance Visualization: Enabling Extreme-Scale Scientific Insight*, chapter The ViSUS Visualization Framework. Chapman & Hall/Crc Computational Science, 2012.

[11] S. J. Plimpton. Fast parallel algorithms for short-range molecular dynamics. 117:1–19, 1995. `http://lammps.sandia.gov/`.

[12] F. Reinders, F. H. Post, and H. J. W. Spoelder. Visualization of time-dependent data using feature tracking and event detection. *The Visual Computer*, 17:55–71, 2001.

[13] W. Schroeder, K. Martin, and B. Lorensen. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, Inc., New York, fourth edition, 2006.

[14] D. Silver and X. Wang. Tracking and visualizing turbulent 3d features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, apr-jun 1997.

[15] D. Silver and X. Wang. Tracking scalar features in unstructured data sets. In *Proceedings of the Visualization '98*, pages 79–86, Oct 1998.

[16] F.-Y. Tzeng and K.-L. Ma. Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 6. IEEE Computer Society, 2005.

[17] W. Widanagamaachchi, P.-T. Bremer, and V. Pascucci. Understanding feature evolution over time using dynamic tracking graphs. *IEEE Transactions on Visualization and Computer Graphics (Submitted)*, 2015.

[18] W. Widanagamaachchi, P.-T. Bremer, C. Sewell, L.-T. Lo, J. Ahrens, and V. Pascucci. Data-parallel halo finding with variable linking lengths. In *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 27–34, Nov 2014.

[19] W. Widanagamaachchi, C. Christensen, V. Pascucci, and P.-T. Bremer. Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In *2012 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 9 –17, oct. 2012.

[20] W. Widanagamaachchi, K. Hammond, L.-T. Lo, B. Wirth, F. Samsel, C. Sewell, J. Ahrens, and V. Pascucci. Visualization and analysis of large-scale atomistic simulations of plasmasurface interactions. In *Proceedings of the 2015 EuroVis - Short Papers*, 2015.