An Interactive Method for Activity Detection Visualization

Li Liu[1], Sedat Ozer[1], Karen Bemis[2], Jay Takle[1], Deborah Silver[1]

[1]Department of Electrical & Computer Engineering, Rutgers University [2]Department of Marine & Coastal Science, Rutgers University

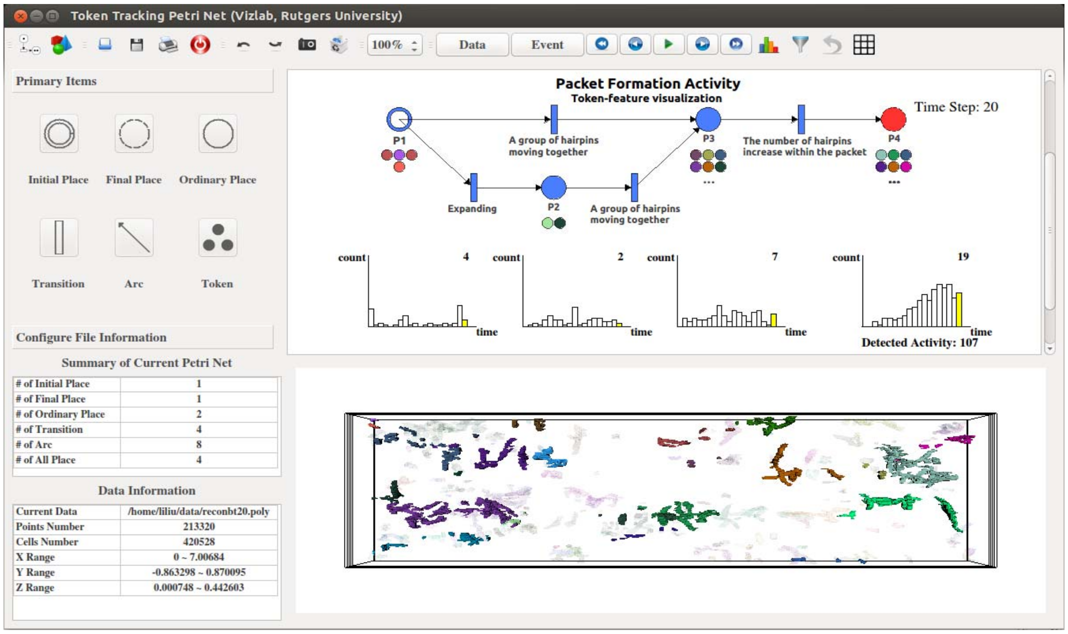


Figure 1: The GUI visualizes activity detection within the 20th time step of a wall bounded turbulence simulation. The simulation is formed of 46 time steps and each time step has the resolution of 384×256×69. The upper part of the "white board" shows a graph-based model of packet formation activity, while the lower part is the visualized data frame in which solid-coloured objects are the features detected as participating in the modelled activity and transparent objects are inactive in the current time step. The tokens under each place (or object state) are the symbolic representation of packets (groups) and so have the same colour with the features they represented. The histograms below visualize the total number of packets in each time step for each place.

Abstract

Visualizing each time step in an activity from a scientific dataset can aid in understanding the data and phenomena. In this work, we present a Graphical User Interface (GUI) that allows scientists to first graphically model an activity, then detect any activities that match the model, and finally visualize the detected activities in time varying scientific data sets. As a graphical and state based interactive approach, an activity detection framework is implemented by our GUI as a tool for modelling, hypothesis-testing and searching for interested activities from the phenomena evolution of the data set. We demonstrate here some features of our GUI: a histogram is used to visualize the number of activities detected as a function of time and to allow the user to focus on a moment in time; a table is used to give details about the activities and the features participating in them; and finally the user is given the ability to click on the screen to bring up 3D images of the overall activity sequence, single time steps of an activity, or individual feature in an activity. We present examples from applications to two different data sets.

Keywords: Graphical user interface, activity detection, data visualization, interactive method, graph-based technique.

1 Introduction

An activity is a complex event (a spatio-temporal pattern) that spans multiple time steps and includes various feature states. Activity detection is the process of searching for all the instances of that pattern in a large data set containing many different types of patterns. In research, the activity detection problem is related to data mining and is treated with different approaches including machine learning and semantic-based techniques. There are many studies on activity detection using these approaches such as those reported in [1, 2, 3, 4]. A successful data mining technique, and thus the activity detection, requires domain knowledge in different forms. For machine learning, the domain knowledge is embedded within the training data; however, it fails to provide an intuitive way to express a hypothesis. In contrast, semantic based approaches do not require training data and allow a domain expert to semantically define an activity and generate a model of the defined activity from its semantic description. As a semantic approach, the graph-based techniques generally use a state based approach in which the various stages of an activity are described as the individual nodes.

In this work, we present a Graphical User Interface (GUI) for visualizing and analyzing both the modelling process and the detected activities in time varying scientific data sets. The fundamental purpose of the GUI is to provide hypothesis-testing through: 1. an interface that facilitates scientific modelling by promoting direct editing of the model by the scientist; 2. a visual connection between the model and the output; and 3. modes of visualization of activities that are scalable to data sets containing millions of features.

2 Implementation

In this implementation of our GUI, we have chosen one particular approach to activity detection "Token Tracking Petri Net" (TTPN) to use as the background processing [1]. The Graphical User Interface (GUI) itself could incorporate a variety of pre-processing and state-based modeling approaches. The GUI is written with standard C++ language and created by Qt with VTK.

3 Primary Features of the GUI

By adding places and transitions to the main board of the GUI, connecting them by arcs, and setting parameters (e.g. place and transition conditions

ll520@scarletmail.rutgers.edu

IEEE Symposium on Large Data Analysis and Visualization 2013
October 13 - 14, Atlanta, Georgia, USA
978-1-4799-1658-0/13/$31.00 ©2013 IEEE

and index, incoming/outgoing arcs, starting time step, and step increment), scientists can create their graph-based activity model manually. They can also load and access to a saved model from text based *config* file. Figure 1 shows the GUI with toolkit, model and data panels.

3.1 Token-feature visualization

The GUI can read the activity list and visualize the detected activities in conjunction with the designed model. In each time step of the data, features that are currently performing the modelled activity will be shown as solid colour while others become transparent. For each active feature, a corresponding token with the same colour will be shown under the place standing for the status the feature is in. By using our GUI in this mode, a scientist can visualize an activity by viewing both the feature tracking results in the data and the token tracking process in the Petri Net simultaneously. Figure 2 demonstrates a step-by-step token-feature visualization example of a merge-split activity spanning three time steps. The simulation data resolution is 128^3 and the data contains 100 time steps.

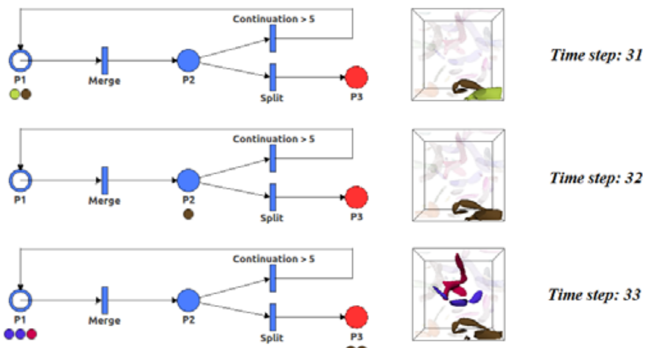


Figure 2: Merge-split activity Visualization. The initial place (empty circle) is starting point for all features and the blue solid circle is the place keeps the merged features. Features that split again within five time steps are shown under the final place (red circle). For each time step, the corresponding features and their location in the data are shown on the right of the model.

Besides visualizing all features performing the modelled activity at current time step, the GUI can also visualize only the active features in a certain state by clicking the place in the model (Figure 3). In this place-based feature visualization, a table listing all token ID of the clicked place (or all feature ID of the data in the corresponding status) will be generated, which provides a tool for scientists to locate and view each feature by choosing token ID from the token list table (Figure 3 and Figure 4). For large data with numerous features at different state, showing the position, shape, and ID of the active features in a certain state of the modelled activity or each individual feature is helpful to analyze the activity.

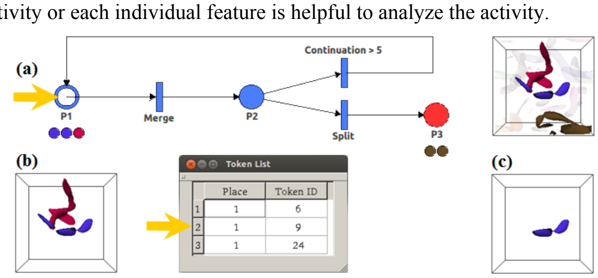


Figure 3: (a) Designed model and the data with active features in the 33[th] time step of the merge-split activity. Click on P1 to see (b) the arising features and a list of token IDs. Click on a token ID from the table of token list and see (c) the corresponding feature in the data.

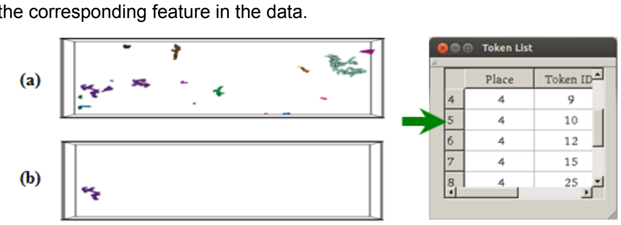
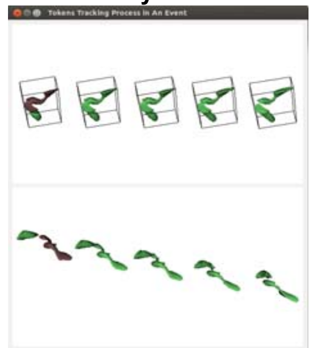


Figure 4: (a) Data with active features of Place 4 in the 20[th] time step of the packet formation activity demonstrated in Figure 1, click on a token ID from the token list table and see (b) the corresponding feature in the data.

3.2 Activity visualization



The GUI can visualize the evolution process of all active features in the modeled activity. Figure 5 visualizes a merge-split activity as multiple stages in the multi-view window.

Figure 5: A multi-view window is divided into two parts: in the upper part, features are extracted from the data frames that span the activity and shown frame by frame; while in the lower part, all active features in this activity are combined into a new data frame and visualized as a whole.

3.3 Histogram of features

The GUI puts a bar chart under each place as a histogram which counts the number of token at each place in every time step. When clicking a bar chart, a histogram window with a series of green bars will pop up. Scientists can view both the data with features and the designed activity model with tokens at any time step by interactively clicking on the specific bar in the histogram window. If a scientist is using the histogram to test the hypothesis that the number of growing packets increases over time (Figure 1), he/she can verify that the packets involved have grown by clicking on the bars of the histogram to bring up a rendering of the data with the involved packets highlighted (Figure 6).

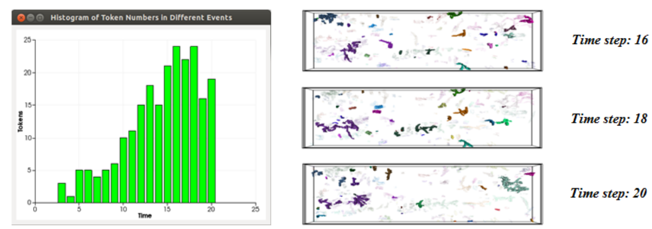


Figure 6: The data with active features of Place 4 in the 16[th], 18[th], and 20[th] time step of the packet-formation activity is visualized by clicking the corresponding bar of the histogram window shown in the left.

4 Conclusion

Visualizing and analyzing time-varying data sets is always a challenge to scientists as the data size and rates increasing. Automated activity detection techniques provide a way for scientists to filter and interact with their data. The GUI presented in this work, as implemented within the activity detection framework TTPN, provides a tool for scientists to design their activity model from their domain knowledge and to visualize the activities hidden in their data. Furthermore, showing active features in a certain state and histogram of features are extensible to large data sets.

Acknowledgements

We gratefully acknowledge the support of DOE #DE-FG02-09ER25977. This work is part of the SciDAC Institute for Ultra-Scale Visualization: http://vis.cs.ucdavis.edu/Ultravis/.

References

[1] S. Ozer, D. Silver, K. Bemis, and P. Martin, “Activity detection for scientific visualization”, Transactions in Visualization and Computer Graphics, in press.

[2] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, “Machine recognition of human activities: A survey”, IEEE Trans. on Circuits and Systems for Video Technology, 18(11): 1473-1488, 2008.

[3] G. Lavee, E. Rivlin, and M. Rudzsky, “Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video” Trans. Sys. Man. Cyber Part C, 39(5): 489-504, 2009

[4] R. Poppe, “A Survey on vision-based human action recognition” Image and Vision Computing, 2010.