# State-based Sensor Data Visualization

Michael Jae-Yoon Chung

## 1 Introduction

rviz [7]

rqt [6]

matlab [4] python [1] r [8]

vizavis [5]

anvil [3]

imovie [2]

Up to the best of my knowledge, I have not yet seen an existing visualization tool that explicitly support data analysis with the purpose of building event detectors. From the robotics community, rviz [7] and rqt_bag [6] are the most related tools that can indirectly help performing such data analysis. When used together, they provide basic data inspection and navigation functionalities such as playback and browse of a recorded sensor data and visualizing snapshot of played back sensor data. However, performing data analysis with the purpose of building event detector with these two tools is a challenging task. First, the user must memorize a summary of visualized sensor data from the multiple time intervals that he/she is interested in. Then, in his/her head compare them to come up with idea for building event detectors. In practice, this is near impossible as the size of sensor data scales up.

Another common approach that many roboticists take is dumping all the sensor data into text file such as .csv and perform analysis with tools like Matlab [4], python [1] or R [8]. In this scenario, the user gains lower level controls for manipulating and visualizing the sensor data, such as feature transformation and visualization of data and feature. However, they now have a full responsibility of developing custom visualization tools, e.g., by scripting.

The Viz-A-Vis project compactly visualizes multiple computer vision features and their aggregations from the large video data (¿ 7500 hours) using 3D visualization techniques. In addition, it provides interactive navigation functionality to help the user inspect the data. While this tool provides a road map of computer vision features which can be useful for understanding events, time intervals comparison functionality which is crucial for building event detectors is not provided.

Other than robotics and ML community, the visualization tools from video editing and annotation fields are inspirational. The ANVIL video annotation tool [3] allows the user to annotation the video in a hierarchical manner. The user can annotate each time frame of the video data with certain state and those annotated state can be later organized in a hierarchical structure. The hierarchical structure hot only makes annotation work more efficient but also help the user to navigate between time frames in more meaningful, organized manner.

Apples iMovie [2] also provides intuitions for organizing and navigating the time-series data. For example, they have features such as loading and

intuitive ways to organize time intervals. They can be loaded and uploaded from main window to quickly inspect Again not really supporting comparison.

## References

[1] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[2] Apple iMoive. `http://www.apple.com/mac/imovie/`, 2014. [Online; accessed 2-24-2014].

[3] M. Kipp. Multimedia annotation, querying and analysis in anvil. *Multimedia information extraction*, 19, 2010.

[4] MATLAB. *version 8.02.0 (R2013b)*. The MathWorks Inc., Natick, Massachusetts, 2013.

[5] M. Romero, J. Summet, J. Stasko, and G. Abowd. Viz-a-vis: Toward visualizing video through computer vision. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1261–1268, 2008.

[6] ROS rqtbag package. `http://wiki.ros.org/rqt_bag`, 2014. [Online; accessed 2-24-2014].

[7] ROS rviz package. `http://wiki.ros.org/rviz`, 2014. [Online; accessed 2-24-2014].

[8] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.

- *Michael Jae-Yoon Chung is with University of Washington, Computer Science & Engineering. E-mail: mjyc@cs.washington.edu.*

*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.*