Pydre: A Python package for driving simulation data reduction

Thomas Kerwin [1]

1 The Ohio State University, United States

DOI: 10.21105/joss.07870

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @fredshone
- @xoolive
- @fxjung

Submitted: 22 January 2025
Published: 10 April 2025

Summary

Driving simulators are complex pieces of equipment that are extremely valuable in experimental studies on driving behavior. Simulation technology is very useful in experimentation when doing something in real life is either too dangerous or too expensive. In driving studies, especially for crash-imminent and driver distraction scenarios, the necessary experimental conditions are often both of these things.

The Python package *Pydre* (pronounced pie-dray), described in this document, provides a cohesive framework for data reduction in the context of real-time, physical driving simulation systems that are used to investigate driving behavior. This software package is intended to be broadly useful for all driving simulation researchers that process data from such mixed and virtual reality driving simulators.

Statement of need

The driving simulators commonly used for the investigation of driving behavior generate a moderate amount of time series data (recorded at 30Hz or more) for each scenario run on the sim. Investigators often want to convert that time series data to discrete objective metrics that describe a specific aspect of how the driver interacted with the vehicle during the drive. These metrics need to be calculated for all participants in a study, and often for multiple scenarios per participant. In this context, we describe the process of converting raw data to a meaningful set of discrete metrics as “data reduction.”

Reducing the raw data from driving simulation experiments has been a task required for the statistical analysis of those experiments for some time (Michelle L Reyes & John D Lee, 2011). Many different individual metrics have been developed over the years as variables of investigatory interest (McManus et al., 2024).

Other researchers have proposed data reduction software packages. Loab et al. describe a Matlab toolkit that performs data reduction (Loeb et al., 2014). The notion of “blocks” in that work is directly tied to the architecture of a previous version of SimCreator (*SimCreator*, 2024), making it less universal. Pawar et al. describe a framework of data reduction similar to Pydre, with regions of interest and metrics (Pawar et al., 2020). However, it lacks the extensible filter system. Neither of these packages are available online in a ready-to-use, runnable form, and both are described in a way that focuses on a specific experimental scenario.

This data reduction is a necessary component of the most common type of driving-data based analysis of participant simulator performance. This work could be accomplished by bespoke processing scripts for a particular project. However, these one-off scripts are often not taken seriously as reusable code and can lead to fragile software projects that can not be easily shared between researchers, even at the same lab. In addition, many approaches using multi-stage processing with multiple programs can lead to intermediate data files and data pipelines that can lead to running code on the incorrect "stage" of the data.

Pydre attempts to avoid the problems above by offering a unified, modular architecture for reducing data from driving simulators into discrete metrics. It is a single application, designed to be used by all projects and share code among them. The modularity allows different users to add new filters or metric-calculating functions in a separate Python file and eventually test and document that code, incorporating it into the general library and allowing easy use for other studies. This modularity and universality is a key goal of the Pydre package and basic functionality is intended to be used with no computer science expertise.

Architecture

In using Pydre to perform data reduction, researchers write a project file describing the data filters, regions of interest (ROIs), and metric functions that will be applied. Filters are applied sequentially to the raw data, then the augmented data files are divided into different ROIs. Finally, metric functions are applied on each ROI, resulting in an output data file that contains one row per ROI in each original drive data file (see Figure 1).

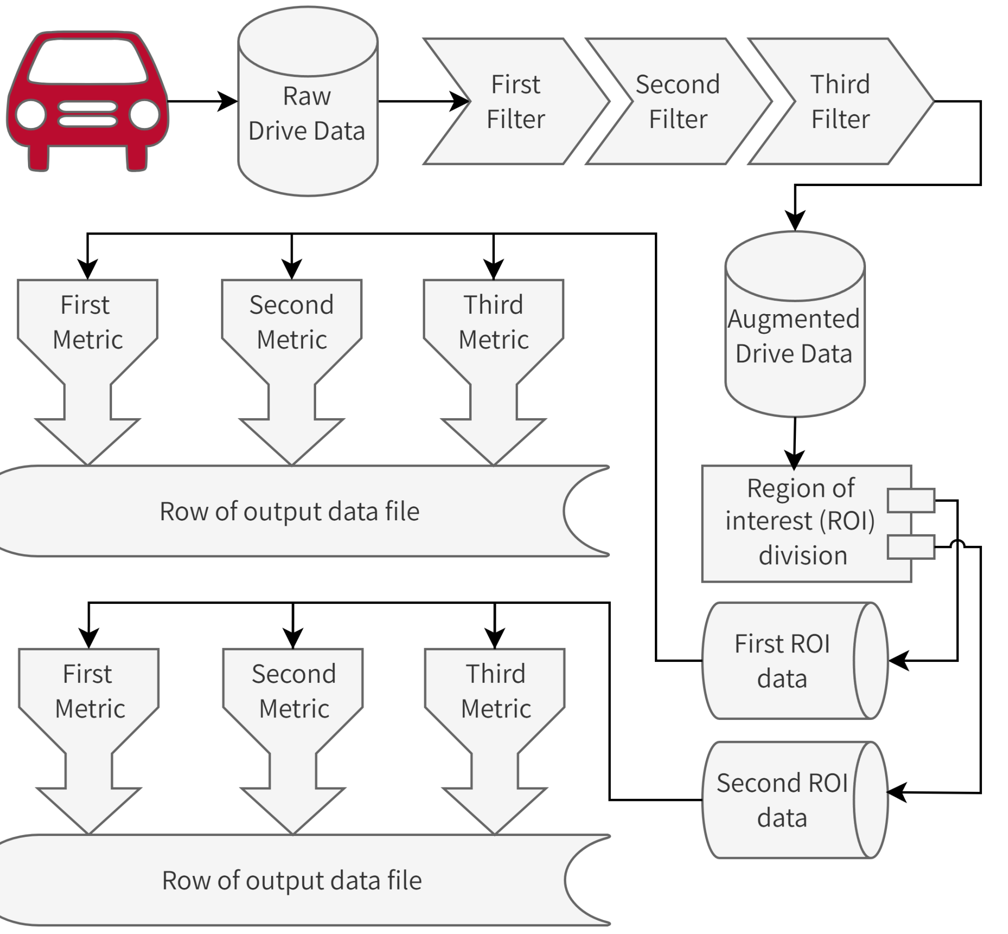

Figure 1: Pydre data pipeline. Data files are filtered individually, broken up in to regions of interest, then metrics are processed for each region.

Raw data files

The raw data files are the output of the driving simulator. They are typically in a tabular format, such as CSV or TSV, and contain time-series data for each participant in the study. Each row in the data file represents a single time step in the simulation, and each column represents a different variable, such as vehicle speed, steering wheel angle, or roadway position, as exported. Each individual data file contains all the data for a single participant driving a single scenario.

Data filters

Data filters augment a raw data file from the sim with additional or changed data. They perform tasks such as merging external data or converting binary marker columns into sequential numbers for ease of later processing. They can also be used to turn raw eye tracking data into fixation numbers.

These filters fulfill the role of data pre-processing in other workflows, but with the Pydre architecture, this is all done in memory with no intermediate files.

Regions of interest

Regions of interest (ROIs) are partitions of the driving data. This is especially useful in repeated-measures experiments, but is also important in removing irrelevant parts of the driving scenario or focusing in on a critical event in the roadway. Two ROI types currently implemented are column ROIs, where the data is partitioned based on flags in the data, often set by scenario scripts or manual switches toggled during the scenario run. Bounding regions, or spatial ROIs, constrain metrics to be calculated in specific regions of the roadway. This can be useful when calculating metrics for a crosswalk zone, a construction zone or any other zone that is constrained in physical space.

Metric functions

Metric functions are those that output the traditional values used to evaluate driving performance in simulation and real-world driving. This includes mean speed, standard deviation of speed, standard deviation of lateral position, steering reversal rate and time above speed limit. The API of metrics provides a standard interface for all items, making it easier for researchers to write new metric functions and reuse metrics in flexible ways.

Metrics are the final output of the data reduction process. They are calculated for each ROI in each data file, and the results are written to a CSV. These metrics files are suitable for loading directly into statistical analysis software and computing ANOVAs or other methods to compare values between conditions or groups. For example, if the ROIs designate the time performing two different tasks (task 1 is ROI 1 and task 2 is ROI 2) and a metric output from Pydre is standard deviation of lateral position, then the output file contains all the information needed to perform a paired t-test to compare the standard deviation of lateral position between the two tasks.

Extensibility

Although there are various real-time interactive driving simulation software systems in use in academic and industry settings, they use very similar tabular, CSV-like, time-series data formats. Pydre was built to ingest data formatted like the output of SimObserver (*SimObserver*, 2024) (a data recorder from Realtime Technologies), but this is a simple TSV interchange data format. Expansion to other driving simulator data formats is ongoing.

Acknowledgements

The authors would like to thank the many undergraduate researchers who have used and provided feedback on previous versions of the software.

References

Loeb, H. S., Seacrist, T., McDonald, C., & Winston, F. (2014). *Simulated Driving Assessment: Case Study for the Development of Drivelab, Extendable Matlab™ Toolbox for Data Reduction of Clinical Driving Simulator Data*. 2014-01-0452. https://doi.org/10.4271/2014-01-0452

McManus, B., Mrug, S., Wagner, W. P., Underhill, A., Pawar, P., Anthony, T., & Stavrinos, D. (2024). Principal components analysis of driving simulator variables in novice drivers. *Transportation Research Part F: Traffic Psychology and Behaviour*, *105*, 257–266. https://doi.org/10.1016/j.trf.2024.05.021

Michelle L Reyes, & John D Lee. (2011). Simulator Data Reduction. In *Handbook of driving simulation for engineering, medicine, and psychology*. CRC Press. https://doi.org/10.1201/b10836-21

Pawar, P., Anthony, T., McManus, B., & Stavrinos, D. (2020). Data Reduction Solution for Driving Simulator. *2020 SoutheastCon*, 1–7. https://doi.org/10.1109/SoutheastCon44009.2020.9249691

SimCreator. (2024). Realtime Technologies. https://www.faac.com/realtime-technologies/products/simcreator/

SimObserver. (2024). Realtime Technologies. https://www.faac.com/realtime-technologies/products/simobserver-pro/