2022 2nd Workshop on Data-Driven and Intelligent Cyber-Physical Systems for Smart Cities Workshop (DI-CPS) | 978-1-6654-7042-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/DI-CPS56137.2022.00010

Intelligent Structuring and Semantic Mapping of Dash Camera Footage and CAN Bus Data

ALEX RICHARDSON, University of Nebraska-Lincoln, USA
KATE SANBORN, The University of Alabama, USA
JONATHAN SPRINKLE, Vanderbilt University, USA

Data sets that contain dash camera and sensor data are essential to the development of autonomous vehicles. Many current methods of constructing these data sets are manually intensive, expensive, and difficult to scale. This paper discusses a method of automating the development of a naturalistic driving data sets that include dash camera footage labelled with information captured from the Controller Area Network (CAN) bus on a vehicle. CAN data can contain IMU data, radar data, etc. First, the dash camera footage and CAN bus data are paired and synchronized using optical flow analysis. Once the footage has been labelled with the telemetric information, one can identify important events in driving behavior by examining the signal conditions. This method is significantly less expensive and more scalable than previous data sets, while providing competitive quality in terms of telemetric data. It could significantly increase the quantity and diversity of driving data sets in the future.

1 INTRODUCTION

Current driving data sets that include dash camera footage and telemetry data suffer from limitations that make gathering data at a large scale prohibitively expensive. This in part due to the need to use devices costing thousands of dollars per car to gather sensor and footage data. This expense limits the scale and geographic diversity of any data sets created since such data gathering is often performed in the researcher's immediate locality. This also puts an onerous burden on labs and researchers with fewer resources than large corporations and universities.

Our solution to these issues is to employ only a standard dash camera and CAN bus reader, which only costs a few hundred dollars per car. CAN data can contain IMU data, radar data, etc. The simplicity and cost reduction of this method allows it to be deployed at a larger extent than previous data sets with a more competitive cost, while still providing a wealth of sensor data. However, this data-gathering method requires that the footage and CAN data files be paired and time-synchronized after data collection. Currently, this is done manually, but that is expensive, time-consuming, difficult to scale, and also onerous to smaller groups with fewer resources.

Hence, this paper shows a method to automate this data processing system by pairing and time-synchronizing dash camera footage and CAN bus data files automatically using optical flow analysis. Our current method is able to automatically pair and time-synchronize 22% of our initial 125 hour data set with no false positives. Work is continuing on improving this metric. Further, this paper also provides techniques for identifying various events of interest using CAN data, such as turns, passes, lead vehicles, and more. This can further expedite data set creation and assist manual event identification and data set structuring.

This paper details the process of creating this data set. Section 2 discusses related studies and approaches. In section 3, the process for preparing the data is explained. Section 4 includes the identified phenomena and the qualifications for each event. In section 5, limitations are discussed. Section 6 identifies potential impact of this work. Section 7 displays the results of this work. Section 8 concludes the paper.

Authors' addresses: Alex Richardson, University of Nebraska-Lincoln, Lincoln, Nebraska, USA, arichardson@huskers.unl.edu; Kate Sanborn, The University of Alabama, Tuscaloosa, Alabama, USA, klsanborn1@crimson.ua.edu; Jonathan Sprinkle, Vanderbilt University, Nashville, Tennessee, USA, sprinkle@acm.org.

2 RELATED WORKS

2.1 Driving Data Sets

There are several existing data sets that include dash camera video with telemetry. For example, the BDD100K data set [9], KITTI data set [3] and Waymo's data set [8] include examples of dash camera footage paired with labeled objects, radar data, and sensor data. However, the data collection in these data sets is expensive - usually requiring custom sensors with technically complicated deployments. This usually restricts these data sets to specific geographic locations. Waymo's data set, for example, is largely restricted to western US cities. Custom sensors include devices such as IMUs, since they do not use CAN data for such telemetry. Even comma2k19's [7] data gathering, which also uses CAN data, requires a dongle that costs $2199 at retail to automatically synchronize footage and telemetry at recording time. All of these data set generation methods are thus prohibitive to scale for many research groups to replicate.

2.2 Synchronizing Video and Sensor Data

Several groups have used optical flow to synchronize dash camera footage with vehicle data. For instance, Fridman and his group [2] correlated the steering angle and vibration of the car with the optical flow of the dash camera footage to synchronize the data. Another group [5] correlated optical flow with turns and acceleration with turns to synchronize their video footage with sensor data.

3 DATA GATHERING AND PREPARATION

3.1 Initial Data Source

The initial data set is 125 hours of video and telemetric data from a Toyota Rav4. This is gathered using a standard dash cam and CAN reader, which only costs a few hundred dollars per car. The dash cam and CAN reader, however, record to separate SD cards, since they function separately. This means that the videos and telemetric data files are not paired, nor time-synchronized - the difference in starting time for a video and its corresponding telemetric file can be up to 2 minutes. This makes it necessary to first determine the appropriate telemetric file for each dash cam video, and then time synchronize it.

978-1-6654-7042-1/22/$31.00 ©2022 IEEE
DOI 10.1109/DI-CPS56137.2022.00010

Currently, this process is done manually. The following sub-sections describe a method for automatically pairing and time-synchronizing a subset of the footage and CAN data. A preliminary optical flow cross-correlation heuristic method is employed that can pair off and time-synchronize approximately 22% of the data from the initial data set with no false positives.

3.2 Cross-correlation methods

Aspects of the CAN data telemetry, such as velocity and yaw, are cross-correlated with the optical flow of a dash camera video as follows.

First, the CAN and associated optical flow signals are obtained:

Name	CAN Equivalent	Optical flow equivalent
Velocity	$v_{can}(t)$	$v_{opt\_total}(t)$
Yaw	$y_{can}(t)$	$y_{opt\_horiz}(t)$

$v_{can}(t)$ and $y_{can}(t)$ are directly derived from the CAN data. $v_{opt\ total}(t)$ and $y_{opt_horiz}(t)$ are signals that are computed using Farneback's [1] optical flow $flow(t)$ as input. Their implementations are Algorithms 1 and 2 respectively. $v_{opt\_total}(t)$ computes a weighted average of total pixel flow using the distance to the center of the flow frame as an inverse weight - movement along the edges of the footage is de-emphasized in favor of movement in the center. This reduces the noise from passing vehicles and obstructions. $y_{opt_horiz}(t)$ is the unweighted average of all horizontal flow.

Algorithm 1 $v_{opt\_total}(t)$

(t)

procedure OPTICALFLOWWEIGHTEDAVERAGE($flow_{t_{length},h,w}$, t) ▷ t_{length} is the time length of the video
  $sum \leftarrow 0$
  $total\_weight \leftarrow 0$
  $base\_distance \leftarrow 0.1$
  for `y, x in flow` **do**
    $x_{flow}, y_{flow} \leftarrow flow_{t,y,x}$
    $length \leftarrow \sqrt{(x_{flow})^2 + (y_{flow})^2}$ ▷ Magnitude of flow at pixel
    $dist \leftarrow \sqrt{(x - (\frac{w}{2.0}))^2 + (y - (\frac{h}{2.0}))^2}$ ▷ Distance to center of frame
    $weight \leftarrow \frac{1.0}{dist + base\_distance}$ ▷ Give change at the center of the frame more weight
    $sum \leftarrow sum + (length * weight)$
    $total\_weight \leftarrow total\_weight + weight$
  end for
  return $\frac{sum}{total\_weight}$
end procedure

The additional *stop* and *log_vel* signals are computed as:

CAN Signal	Optical Signal
$stop_{can}(t) = \begin{cases} 1, & \text{if } v_{can}(t) \leq 1 \\ 0, & \text{else} \end{cases}$	$stop_{opt}(t) = \begin{cases} 1 & v_{opt\_total}(t) \leq 1 \\ 0 & \text{else} \end{cases}$
$log\_vel_{can}(t) = \log(v_{can}(t))$	$log\_vel_{opt}(t) = \log(v_{opt\_total}(t))$

From these signals, the following respective signals from the CAN and optical flow are cross-correlated together:

Algorithm 2 $y_{opt\_horiz}(t)$

(t)

procedure OPTICALFLOWYAW($flow_{t_{length},h,w}$, t)
  $sum \leftarrow 0$
  $count \leftarrow (h * w)$
  for `y, x in flow` **do**
    $x_{flow} \leftarrow flow_{t,y,x}$
    $sum \leftarrow sum + x_{flow}$
  end for
  return $\frac{sum}{count}$
end procedure

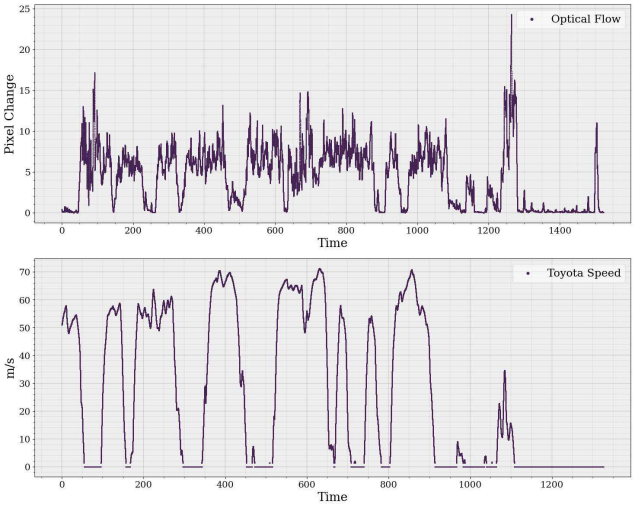

Fig. 1. Raw velocity pair graphs as a baseline comparison

Signal Name	CAN Signal	Optical Signal
Stop Signal	$stop_{can}(t)$	$stop_{opt}(t)$
Log-velocity Signal	$log\_vel_{can}(t)$	$log\_vel_{opt}(t)$
Yaw Signal	$y_{can}(t)$	$y_{opt\_horiz}(t)$

The resulting properties of each cross-correlation are listed as such. Shifts are in seconds:

Signal Name	Correlation Coefficient	Shift
Stop Signal	c_{stop}	s_{stop}
Log-velocity Signal	c_{logv}	s_{logv}
Yaw Signal	c_{yaw}	s_{yaw}

One can see the comparisons between these optical flow signals and CAN signals in Figures 1 - 4 on a sample drive.

3.3 Pairing CAN data with Dash camera videos

For the purposes of pairing CAN data with videos, all possible combinations of CAN files and dash camera files are considered, and the cross-correlations are computed as described. The resulting combinations and computed cross-correlations are then sorted from highest-to-lowest for c_{logv}, which serves as a rough measure of confidence. This list of combinations is then iterated, from the beginning to the end, with each combination being flagged as a valid pairing if the following conditions are met:

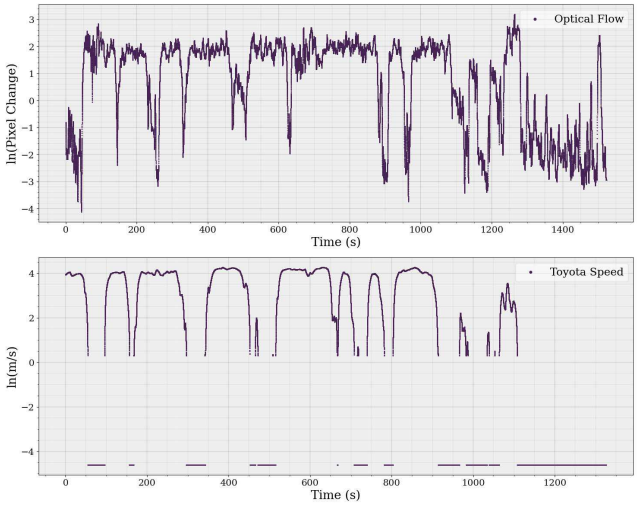

Fig. 2. Log-velocity pair graphs

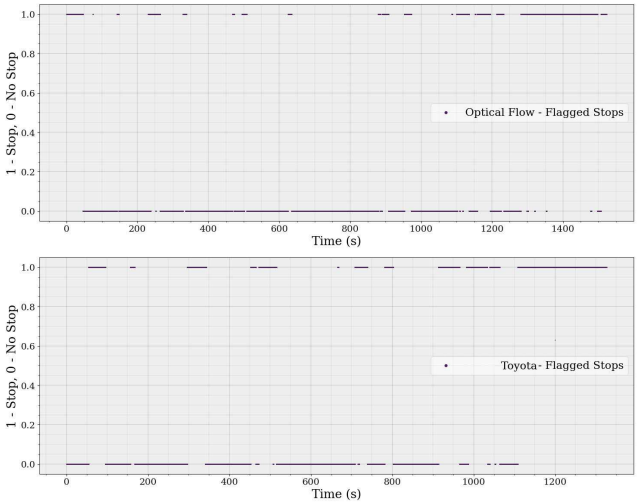

Fig. 3. Stop method pair graphs

(1) Neither the CAN file or the dash camera file have already been marked as part of another pairing.
(2) The CAN timestamp is within 15 hours of the dash camera datestamp, which is translated to a timestamp of noon CST of the date.
(3) $c_{logv} > 0.2$
(4) $|s_{logv} - s_{yaw}| < 5$ and $|s_{logv} - s_{stop}| < 5$

3.4 Time synchronization of CAN data with Dash camera videos

Once the CAN data files have been paired with the appropriate videos, the CAN data and the video must be time-synchronized, i.e., the start-time of the CAN data must correspond with the first frame of the video. Using the cross-correlation methods already discussed, the appropriate time-shift is selected by averaging the shifts from the three cross-correlations.

The CAN data and video are then appropriately trimmed so that they are of the same temporal length.

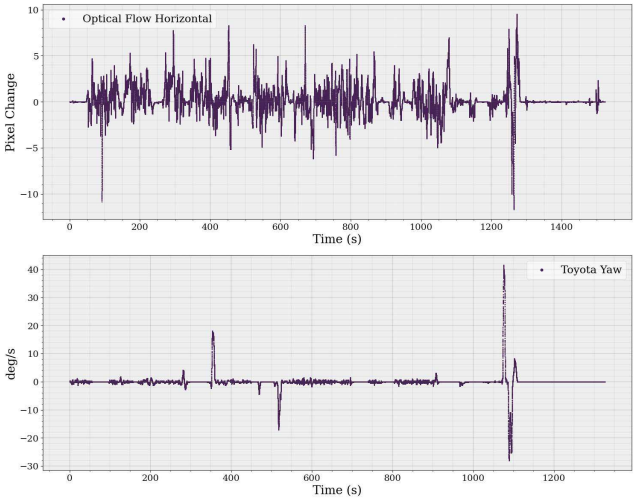

Fig. 4. Raw yaw method pair graphs

4 IDENTIFYING EVENTS

Define the distance to the lead vehicle $s_{lead} = \{b, \emptyset\}$ where $b \in \mathbb{R}+$ where this value may be undefined at certain times shown by $\emptyset$. Likewise we define side radar the distance measured by one of sixteen radar traces as $tr_{number} = \{b, \emptyset\}$ where $b \in \mathbb{N}$ where this value may be undefined at certain times shown by $\emptyset$. The blind spot monitor is defined as $approach = \{0, 1\}$ where 1 indicates a blind spot warning triggered on a given side of the vehicle. The lead vehicle relative speed is defined as $v_{lead} = \{b, \emptyset\}$ where $b \in \mathbb{R}$ where this value may be undefined at certain times shown by $\emptyset$. The cruise control state is defined as $c_{state} = \{0, 1\}$ where 1 indicates that the vehicle cruise control is active. Vehicle acceleration is defined as $a \in \mathbb{R}$. Define the turn signal state as $turn_{state} = \{1, 2, 3\}$ where 1 indicates an active left turn signal, 2 indicates an active right turn signal, and 3 indicates no turn signal. Define the steering angle as $\theta \in \mathbb{R}$. Define brake state $b = 0, 1$ where 1 indicates that the brake pedal is being pressed.

If a signal $s(t)$ is true for some time T, then we can say that $\exists t_0$ s.t. $s(t) = 1$ for all $t \in [t_0, t_0 + T]$. In the following subsections, these signal definitions will be used to describe the conditions to identify various examples of driving behavior with signal temporal logic.

4.1 Lead Vehicle

For identifying sections of chunks of CAN data and footage where the vehicle is following another vehicle, each time interval of CAN data where the recorded lead distance is less than 250 meters is labeled. The 250 meter threshold is used since the car telemetry records the absence of a lead vehicle as a lead distance of greater than 250 meters. These conditions are demonstrated by the following

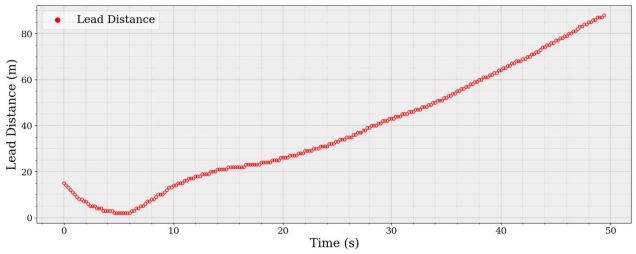

Fig. 5. Lead Distance

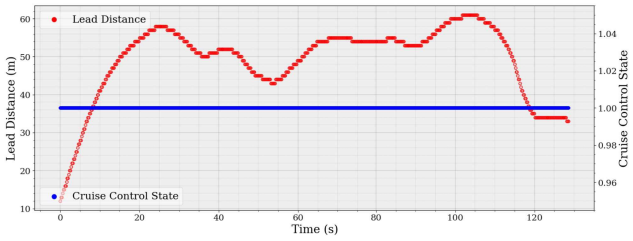

Fig. 6. Lead Vehicle with Cruise Control

expression:

$$\mathbf{F}(\mathbf{G}(s_{lead}(t) < 250))$$

The lead distance is grouped into chunks by discontinuities in the data. A jump in the lead data indicates that a lead vehicle moves in front of the car or moves out of the way of the car. Each of these chunks therefore represents a period where the car was behind one vehicle. Each chunk of lead data is then output as a video clip. Figure 5 demonstrates data points that meet these conditions.

4.2 Lead Vehicle with Cruise Control

Identifying moments with a lead vehicle present and when the cruise control is active involves looking at the lead distance and active cruise control signals in the CAN bus data. The lead distance data is filtered to remove values that do not detect any vehicles. Radar traces are also used to fill some gaps that were present in the data. Chunks of lead distance data representing the same lead vehicle are created. These chunks are further divided into clips based on whether or not cruise control was active. These smaller clips that include a lead vehicle and active cruise control are added to the list of desired time intervals. These conditions are also represented as follows:

$\mathbf{F}(\mathbf{G}((s_{lead}(t) < 250) \wedge (c_{state}(t) = 1))$

Then, using these intervals, the appropriate clips of video and CAN bus data are created. An example of data that matches these conditions is shown in Figure 6.

4.3 Short Lead Vehicle

To identify clips where a lead vehicle is present for five seconds or less, the first step is to filter the lead distance by removing unlikely values and filling in gaps with radar traces. Next, the lead distance data is divided into chunks where each chunk represents a distinct vehicle. Only chunks with lengths less than or equal to five seconds

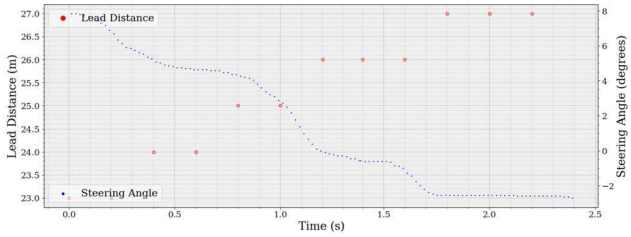

Fig. 7. Short Lead Signals

and greater than one second are kept. This interval length is used to find short following sections while ignoring clips less than a second that often represent false positives. Within these chunks, the steering angle is checked. If the time interval for the chunk has a steering angle greater than fifteen degrees, this signifies that the car is likely turning left or right. Any lead data collected on this turn is likely to be a parked car or other cars that are not lead vehicles. Therefore, time intervals with these large steering angles are thrown out. These conditions can also be expressed as follows:

$\mathbf{F}(\mathbf{G}_{[0,\tau]}(s_{lead}(t) < 250 \wedge \theta(t) <= 15)), 1 < \tau <= 5$

The remaining short time intervals with lead vehicles are translated into video clips with the corresponding CAN bus data. An example of data that fits these conditions is shown in Figure 7.

4.4 Long Lead Vehicle

For the purposes of this project, a long period of time where a lead vehicle is present is defined as having a length of at least thirty seconds. The signal used to identify these events is the lead distance. Each continuous section of lead distance represents following a lead vehicle for a period of time. Only the chunks that are thirty seconds or more are kept. Some of the dash camera footage contains instances where the vehicle is parked with something in front of it for longer than thirty seconds. This was incorrectly being labeled as following a lead vehicle. To resolve this issue, only instances with a lead distance of greater than one meter are considered. When following a car while driving, a safe following distance should be greater than one meter. Anything with one meter or less lead distance can therefore be removed. These conditions can also be expressed as follows:

$\mathbf{F}(\mathbf{G}_{[0,\tau]}(1 < s_{lead}(t) < 250)), 30 < \tau < \infty$

The remaining time segments are output as clips of dash camera footage and the CSV file for the CAN bus data. An example of data that matches these conditions is shown in Figure 8.

4.5 Passes

Instances where another car passes the vehicle have also been identified. The first signal used is lead distance. When a decrease in lead distance is located, this indicates a car moving in front of the vehicle. This time (t = 0 s) is used as a reference for checking other signals. The next signal examined is the blind spot monitors. These indicate whether or not a car is approaching on the left or right side of the vehicle. The script checks ten seconds before (-10 s <t <0 s) to ensure that an approaching car has been detected before the decrease in lead distance. The radar traces are also examined ten

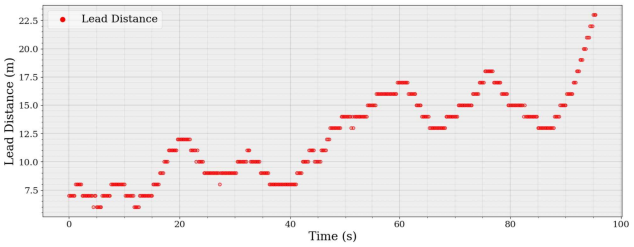

Fig. 8. Long Lead

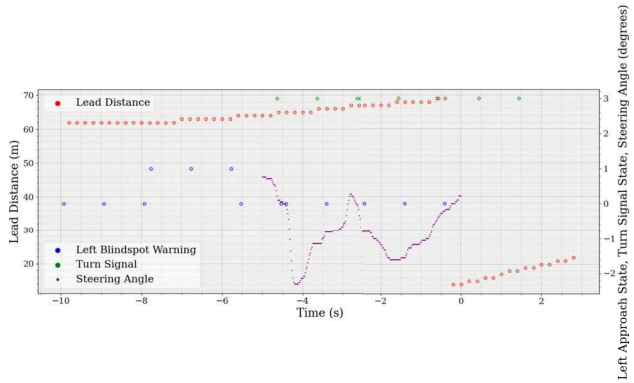

Fig. 9. Example Pass Signals

seconds before and three seconds after (-10 s <t <3 s) to make sure that a car approaching on the left appears in front on the left or a car approaching on the right appears in front on the right. Using these signals is not enough to identify a passing car. In some cases, a lane change satisfies all of the criteria. Turn signal and steering angle signals are examined to filter out the lane changes. If a turn signal is active five seconds before (-5 s <t <1.5 s) or the steering angle exceeds three degrees [4] five seconds before (-5 s <t <0 s) then the case is ignored. The remaining cases that are not lane changes are output as video clips and the corresponding sections of the CAN bus data. All the identified time intervals are determined based on the driving data that has been collected. Through experimentation, these conditions have been showed to yield accurate and comprehensive results. These conditions can be expressed as follows:

$$\begin{aligned} \mathbf{F}((s_{lead}(0) \ll s_{lead}(0-\Delta t)) \quad & \wedge \\ \mathbf{F}_{[-10,0]}(approach(t)=1) \quad & \wedge \\ \mathbf{F}_{[-10,3]}(tr_{number}(t)<25) \quad & \wedge \\ \mathbf{G}_{[-5,1.5]}(turn_{state}(t)=3)) \quad & \wedge \\ \mathbf{G}_{[-5,0]}(\theta(t)<=3) \quad & \end{aligned} \tag{1}$$

An example of data that matches these conditions is shown in Figure 9. The radar traces are not included in this figure to simplify the graph.

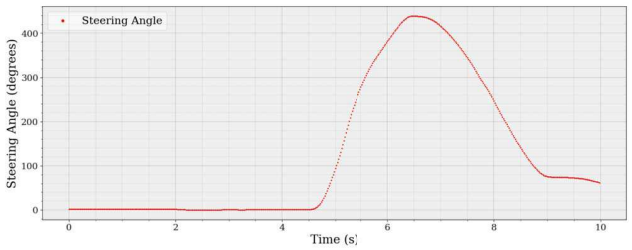

Fig. 10. Turn

4.6 Turns

For identifying sections of chunks of CAN data and footage where the vehicle is turning, each time interval of CAN data where the absolute value of the recorded steering angle is greater than 100 degrees is added. It is unclear why the data analysis yields 100 degrees as the optimal threshold. These conditions can also be expressed as follows:

$\mathbf{F}(\mathbf{G}(\theta(t) >= 100))$

Figure 10 demonstrates data that meets these conditions.

4.7 Braking Events

Braking events are identified and classified from the dash camera footage. For this project, a hard brake is defined as a deceleration of 3.5 m/s^2 or more for at least half a second. A medium brake is defined as a deceleration of 2 m/s^2 or more for at least half a second. Any smaller deceleration is defined as a soft brake. These acceleration thresholds are defined based on [6]. The main signals examined to identify braking are longitudinal acceleration and brake pressed. Acceleration is filtered to remove any positive values and any time where the brake is not pressed. The minimum acceleration is calculated for the remaining intervals. If the acceleration is within one of the thresholds for at least half of a second, the braking event is labeled with the corresponding category. The half second threshold ensures that the car was within the range for a significant amount of time and can be placed in the most representative category. The hard brake conditions are as follows:

$\mathbf{F}(\mathbf{G}_{[0,\tau]}(b(t)=1) \wedge \mathbf{F}_{[0,\tau]}(\mathbf{G}_{[0,0.5]}(a(t) <= -3.5))), \tau >= 0.5$

Data that matches this is displayed in Figure 11.

The medium brake conditions are these:

$\mathbf{F}(\mathbf{G}_{[0,\tau]}(b(t)=1) \wedge \mathbf{F}_{[0,\tau]}(\mathbf{G}_{[0,0.5]}(-3.5 < a(t) <= -2))), \tau >= 0.5$

Data that matches this is displayed in Figure 12.

The soft brake conditions are these:

$\mathbf{F}(\mathbf{G}_{[0,\tau]}(b(t)=1) \wedge \mathbf{F}_{[0,\tau]}(\mathbf{G}_{[0,0.5]}(-2 < a(t) < 0))), \tau >= 0.5$

Data that matches this is displayed in Figure 13. The names of the outputted videos and CAN data files reflect the braking intensity categories.

5 LIMITATIONS

Our approach for pairing, synchronization, and event identification comes with considerable limitations.

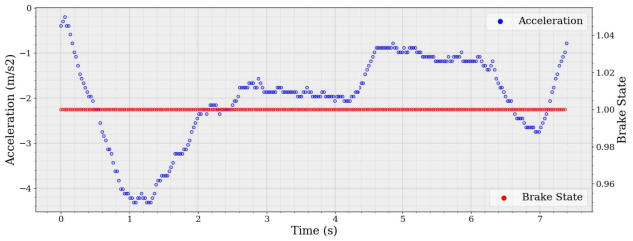

Fig. 11. Hard Brake

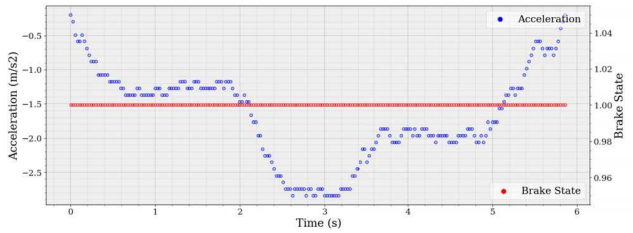

Fig. 12. Medium Brake

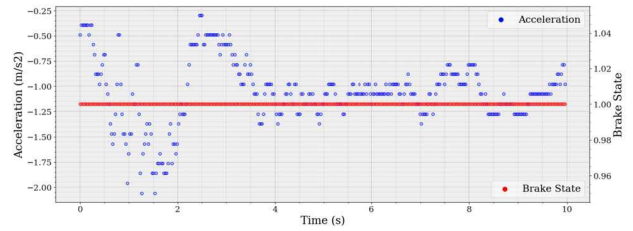

Fig. 13. Soft Brake

Our approach for automated pairing and time-synchronization has room for improvement. Currently, our automated pairing is only able to pair approximately 22% of the data from our initial data set. We hypothesize that this is likely to be consistent across other data sets as well. The main contributing factor to this low yield rate is the low correlation between our computed optical flow and the velocity and yaw. This results in discrepancy between the computed cross-correlations, and thus lower yield rates.

Identification of events within the data set has limitations as well. First, our initial data set had few examples of certain events, such as hard braking. This makes it more difficult to ascertain the replicability and robustness of the methods used to obtain these events. Further, our methods make assumptions about "normal" driving behavior, and how certain signals from telemetry could thus be reflective of certain events. Abnormal events inconsistent with typical every day driving may possibly be spuriously included in these automatically annotated event data sets.

6 POTENTIAL IMPACT

In the dash camera footage, license plate numbers and images of people are sometimes visible. Also, some videos start and end in residential areas. Both of these present privacy concerns. To address these issues, the quality of the videos is reduced so that license plate numbers are not visible. Furthermore, given the speed of the vehicle, it is not possible to identify any person included in the video. Additionally, videos will be clipped to only include roads that do not present privacy concerns.

7 RESULTS

This table includes the results of the pairing and synchronization process for the data set:[1]

Category	Time
Total Paired and Synchronized Data	28:14:04
Lead Vehicle	14:17:05
Lead Vehicle with Cruise Control	00:26:10
Short Lead Vehicle	00:14:33
Long Lead Vehicle	12:10:12
Passes	00:02:45
Turns	01:41:10
Hard Brake	00:01:19
Medium Brake	01:02:19
Soft Brake	03:32:08

8 CONCLUSION AND FUTURE WORK

This paper presents methods of pairing and synchronizing vehicle dash camera footage and corresponding CAN bus data by cross-correlating the sensor data with the optical flow of the footage. This produced many successful synchronized pairs that were then used to automate the identification of several hours of meaningful driving events. Notably, up to 100,000 hours of data are expected from the research setup by the conclusion of 2022, making it imperative to manage the complexity of event detection, and to search for and extract relevant activities that may precipitate or follow events, for purposes of training new algorithms on full data sets. As more data continues to be collected, these methods promise to accelerate the processing of high quality and quantity data.

Currently, our pairing and time-synchronization method involves cross-correlation of optical flow and CAN data after applying simple transformations to both. This simplicity currently restricts our pairing rate to 22% for our data set. Two complimentary approaches are currently being investigated to improve this rate. The first approach is a machine learning-based approach to enhance estimation of velocity and yaw from the footage. The second approach is to exploit the preserved temporal ordering of the dash cam videos and CAN telemetric files to apply a dynamic programming approach to the already described cross-correlation pairing method.

REFERENCES

[1] Gunnar Farnebäck. 2003. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*. Springer, 363–370.
[2] Lex Fridman, Daniel E. Brown, William Angell, Irman Abdić, Bryan Reimer, and Hae Young Noh. 2016. Automated synchronization of driving data using vibration and steering events. *Pattern Recognition Letters* 75 (2016), 9–15. https://doi.org/10.1016/j.patrec.2016.02.011
[3] A Geiger, P Lenz, C Stiller, and R Urtasun. 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237. https://doi.org/10.1177/0278364913491297

[1]These numbers are included to provide a general idea of the scope of the event identification. Footage analyzed in this paper may not match future data set release.

[4] Hongyu Hu, Zhenhai Gao, Ziwen Yu, and Yiteng Sun. 2017. An experimental driving simulator study of unintentional lane departure. *Advances in Mechanical Engineering* 9, 10 (2017), 1–8. https://doi.org/10.1177/1687814017726290

[5] Yosuke Ishiwatari, Takahiro Otsuka, Masahiro Abukawa, and Hiroshi Mineno. 2020. A Data Synchronization Method for a Vehicle Using Multimodal Data Features. *International Journal of Informatics Society* 11, 3 (2020), 135–147.

[6] Jwan Kamla, Tony Parry, and Andrew Dawson. 2019. Analysing truck harsh braking incidents to study roundabout accident risk. *Accident Analysis and Prevention* 122 (2019), 365–377. https://doi.org/10.1016/j.aap.2018.04.031

[7] Harald Schafer, Eder Santana, Andrew Haden, and Riccardo Biasini. 2018. A Commute in Data: The comma2k19 Dataset. arXiv:cs.RO/1812.05752

[8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2446–2454.

[9] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). https://doi.org/10.1109/cvpr42600.2020.00271