DDD17: End-To-End DAVIS Driving Dataset

Jonathan Binas [1] **Daniel Neil** [1] **Shih-Chii Liu** [1] **Tobi Delbruck** [1]

Abstract

Event cameras such as dynamic vision sensors (DVS) and dynamic and active-pixel vision sensors (DAVIS) can supplement other autonomous driving sensors by providing a concurrent stream of standard active pixel sensor (APS) images and DVS temporal contrast events. The APS stream is a sequence of standard grayscale image sensor frames. The DVS events represent brightness changes. They have dynamic range of $>$120 dB and effective frame rates $>$1 kHz with data rates comparable to 30 fps (frames/second) image sensors. To overcome some of the limitations of current image acquisition technology, we investigate in this work the use of the combined DVS and APS streams in end-to-end driving applications. We provide DDD17, the first open dataset of annotated DAVIS driving recordings. DDD17 has 12h of a 346x260 pixel DAVIS sensor recording highway and city driving in daytime, evening, night, dry and wet weather conditions, along with vehicle speed, GPS position, etc., and driver steering, throttle, and brake captured from the car's on-board diagnostics interface. As an example application, we performed a very preliminary end-to-end learning study of using a convolutional neural network that is trained to predict the instantaneous steering angle from DVS and APS visual data. We provide networks that compute the steering angle using a CNN and a network that includes a small recurrent neural network at the output of the CNN.

1. Introduction

The rapid improvement of machine learning and computer vision systems has spurred the development of self driving cars, which have already covered millions of kilometers in real world scenarios. It appears that the development of processing technology and algorithms currently advances at greater speed than the development of sensing hardware for capturing the necessary information from the surroundings of the vehicle, such as obstacles, traffic, marks, and signs. Automotive image sensors are being intensively developed to deal with the conflicting requirements for low cost, high dynamic range, high sensitivity, and resistance to artifacts from flickering light sources such as LED traffic signs and car taillights. Operation under bad weather or lighting conditions is a primary requirement for automotive self driving or automatic driver assistance systems (**ADAS**), but current ADAS sensors and systems still face many problems compared to human driver performance under challenging situations. Since event cameras have been proposed as possible ADAS sensors (Posch et al., 2014), we collected data to study the use of an event camera to augment conventional imager technology.

Rather than providing frame-based video as output, the event camera dynamic vision sensor (**DVS**) detects local changes in the brightness of individual pixels and asynchronously outputs those changes at the time of the changes (Lichtsteiner et al., 2008; Posch et al., 2014). Thus, only parts of the scene that change produce data, lowering the output data rate, increasing the temporal resolution and reducing the latency in comparison to frame-based systems, since changes in pixel brightness are streamed out of the camera as they occur. The local instantaneous gain control increases usability under uncontrolled lighting conditions. The higher temporal resolution and limited data rate makes the DVS well suited for autonomous driving applications, where both latency and power consumption are important. A dynamic and active-pixel vision sensor (**DAVIS**) has pixels that concurrently output DVS events and standard image sensor intensity frames (Brandli et al., 2014).

Recent studies have shown the utility of using DVS in data-driven convolutional neural network (CNN) real time applications (Moeys et al., 2016; Lungu et al., 2017). In these applications, DVS input frames consisted of a 2D histogram image of a constant number of a few thousand DVS events. Pixels of this histogram start out gray and are drawn whiter or blacker as they accumulate ON and OFF DVS events. Because the DVS event rate is proportional to the rate of change of brightness (i.e. scene reflectance

[1]Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. Correspondence to: Jonathan Binas <jbinas@ini.uzh.ch>, Tobi Delbruck <tobi@ini.uzh.ch>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

(Lichtsteiner et al., 2008), the CNN frame rate is variable, ranging from about 1 fps up to 1 kfps. Moeys et al. (2016) showed that combining the standard image sensor frames from the sensor with the DVS frames resulted in higher accuracy and lower average reaction time latency. Here we extend this work to real world driving in the first published end-to-end (**E2E**) dataset of DVS or DAVIS driving data.

2. Davis Driving Dataset 2017 (DDD17)

We collected data from Swiss and German road driving. The DDD17 dataset will be hosted at sensors.ini.uzh.ch/databases. For review, the dataset link is provided [1] . The link will be published on paper acceptance.

2.1. DAVIS data

Visual data was captured using a DAVIS346B prototype camera, containing a DAVIS APS+DVS camera, such that event-based and traditional frame-based data could be recorded at the same time, through the same optics. The camera resolution is 346×260 pixels. The camera architecture is similar to Brandli et al. (2014), but the sensor has 2.1X more pixels and includes on-chip column parallel analog to digital converters (ADCs) for frame-based APS output up to 50 fps. The DAVIS346B also has optimized photodiodes with microlenses that increase fill factor and reduce dark current, thereby improving operation at low light intensities. A fixed focal length lens (C-mount, 6mm) was used for all recordings, providing a horizontal field of view of 56°. The aperture was set manually, depending on lighting conditions. The APS frame rate was dependent through exposure duration on lighting conditions to a value between 10 fps and 50 fps; in some recordings it varied depending on the auto-exposure duration algorithm. The frames were captured using the DAVIS global shutter mode to minimize motion artifacts. The camera was mounted using a glass suction tripod mount behind the windshield, just below the rear mirror, and aligned to point to the center of the hood. Markers on the car hood were used to align the camera for each recording session. A polarization filter was used in some of the recordings to reduce windshield and hood glare. The camera was powered and connected to a laptop computer by high speed USB 2.0. The raw data was read out using inilabs cAER software[2] and streamed to the custom recording framework described in Sec. 2.3 for further processing.

[1]DDD17 web page
[2]cAER support

2.2. Vehicle control and diagnostic data

Data was acquired using a rented Ford Mondeo MK 3 European Model. We used the $130 OpenXC Ford Reference vehicle interface, that plugs into the passenger compartment OBDII port, to read out control and diagnostic data from the car's CAN bus. The vehicle interface connects to a host USB port[3].

The vehicle interface was programmed with the vendor-provided firmware for the Ford Mondeo MK 3 car model and read out using the OpenXC python library[4]. The raw data was passed to the custom recording software described in Sec. 2.3. The following quantities were read out at rates of about 10 Hz each. Possible targets for experiments in E2E learning are in boldface.

- **steering wheel angle** (degrees, up to 720°)
- **accelerator pedal position** (% pressed),
- **brake pedal status** (pressed/not pressed),
- engine speed (rpm),
- vehicle speed (km/h),
- latitude,
- longitude,
- headlamp status (on/off),
- high beam status (on/off),
- windshield wiper status (on/off),
- odometer (km),
- torque at transmission,
- transmission gear position (gear no.),
- fuel consumed since restart,
- fuel level (%),
- ignition status,
- parking brake status (on/off).

2.3. Recording and viewing software

A python software framework [5] for recording, viewing, and exporting the data was created for the main purpose of combining and synchronizing the data from the different input devices and storing it in a standardized file format. In particular, since the APS frames and DVS data are microsecond time-stamped on the camera using its own clock, whereas the data provided by the vehicle interface is not, both data streams were augmented with the millisecond system time of the recording computer, which could then be used for synchronization. The computer time was synchronized to a standard time server before recordings. Both streams were processed by individual processes to ensure timely addition of the timestamps. With the vehicle interface streaming data at rates of only around 10 Hz per recorded variable, such off-device time-stamping is jus-

[3]OpenXC vehicle interface
[4]OpenXC getting started guide
[5]ddd17-utils

tified. The data was stored in HDF5 format, for which widely used libraries for various environments exist. Each data type (e.g. DVS events, steering wheel angle, vehicle speed...) was stored in a separate container, each containing one container for the system timestamp and one for the data. In this way, the system timestamp can be used for fast indexing and for synchronizing the data when reading. With data being provided at irregular intervals by the recording devices, each data type was stored in an event-driven fashion, such that different containers contain different numbers of samples. The DAVIS data was stored in its native cAER AER-DAT3.1 format[6] in each HDF5 container.

In addition to the recording framework, a python viewer `view.py` visualizes the recorded DAVIS data together with selected vehicle data such as the steering angle or speed (Fig. 1). The script `export.py` exports the data into frames for preparing data for further processing by machine learning algorithms.

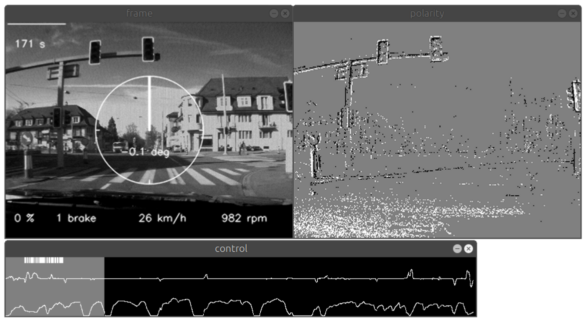

Figure 1. Example scenario visualized by the recording file viewer. The top panels show the DAVIS frames (left; overlaid with some driving data) and events (right), the bottom panel shows a progress bar as well as visualizations of different vehicle data (headlamp status at the top, steering angle in the middle, speed at the bottom).

3. Recorded data

In total, over 12 h of data were recorded under various weather, driving, road, and lighting conditions on six consecutive days, covering over 1000 km of different types of roads in Switzerland and Germany. Recordings were started and stopped manually and typically have durations of between a minute and an hour. The resulting recordings are summarized in Table 1. Fig. 2 shows the distributions of several recorded variables over the whole dataset. Steering angles are dominated by straight driving and small deviations of $\pm 10°$. Speed is uniformly distributed over the range 0-160 km/h. The automatically controlled headlight is on about half the time, indicating a substantial fraction of the data was captured in low-light conditions.

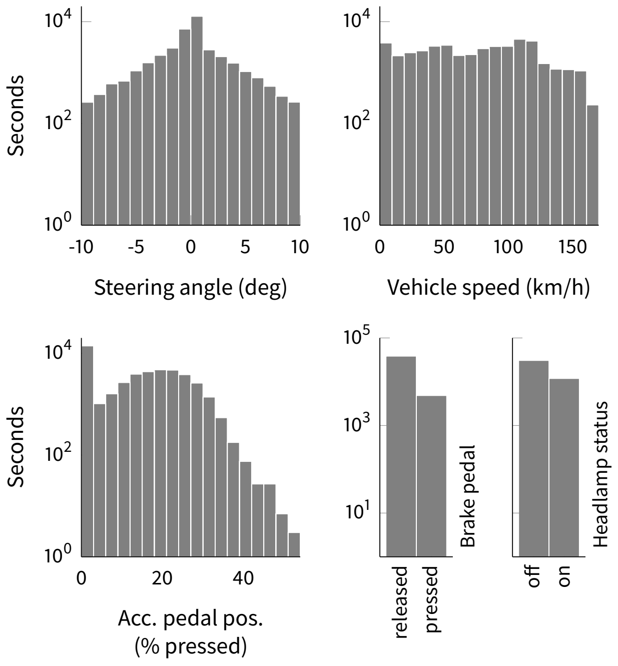

Figure 2. Statistical distribution of various recorded signals.

4. Experiments: Steering prediction network

E2E learning of a control model is an initial approach for self-driving applications, since it eliminates the need for tedious hand-labeling of the data or features – a task which is prohibitive in the face of the enormous amounts of data acquired by today's vehicles (Bojarski et al., 2016). E2E has clear limitations, since it cannot predict user intentions and the dataset tends to be very unbalanced. Nevertheless, under some conditions such as highway driving or driving along roads without turns onto other roads or unpredictable user actions, it can be used to study the quality of recorded data.

We trained simple steering prediction networks. These networks take input APS and/or DVS data and attempt to predict the instantaneous steering wheel angle. They are inspired by LeCun's early work (LeCun et al., 2005), the seminal open dataset from comma.ai (Santana & Hotz, 2016), and by recent Nvidia (Bojarski et al., 2016) and unpublished VW studies.

Our results compared the steering prediction accuracy of using pure APS, pure DVS, and combined APS and DVS data. Our study should be regarded as a preliminary study mainly done to validate the usability of the data and associated scripts. We only had time to train on two recordings (1487858093 and 1487433587 in Table 1). Work is ongo-

[6]inilabs file formats

File(.hdf5)	Scene	Cond.	Dur. (s)	GB
1487339175	cty	wet	347	2.8
1487349453	campus	dark	192	1.7
1487350455	fwy	ngt, rain	1404	11.2
1487354030	cty	ngt, wet	377	3
1487354811	cty	ngt, wet	190	1.4
1487355025	cty	ngt, wet	57	0.4
1487355090	cty, hwy	ngt, wet	984	5.9
1487356509	fwy	ngt, wet	2233	12.4
1487417411	fwy	day	2096	18.2
1487419513	fwy	day	1976	18.3
1487424147	m. fwy	day	3040	30.3
1487427200	fwy	day	1947	17.6
1487430438	fwy	day	3135	26.2
1487433587	fwy+cty	ngt	2355	18.5
1487593224	hwy	day	586	5.3
1487594667	fwy	day	2985	29.7
1487597945	cty	evening	50	0.5
1487598202	fwy	day	1882	15.1
1487600962	fwy	day	2143	15.1
1487608147	fwy	evening	1208	9
1487609463	fwy	evening	1458	6.3
1487778564	campus	day	101	1.1
1487779465	cty+hwy	day	1170	11.2
1487781509	campus	evening	127	0.6
1487782014	cty+hwy	evening	1118	7.3
1487839456	cty	day, sun	406	5.7
1487842276	cty	day, sun	625	9.5
1487844247	cty	day, sun	523	7.5
1487846842	twn+hwy	day, sun	1799	20.6
1487849151	twn	day, sun	429	5.5
1487849663	twn+hwy	day, sun	2863	34.7
1487856408	twn	day, sun	817	13.2
1487857941	twn	day, sun	99	1.4
1487858093	cty	day, sun	2421	34.7
1487860613	cty	day, sun	1065	17.4
1487864316	cty+fwy	evening	1087	12.9

Table 1. Summary of the acquired data. Keys: hwy=highway, fwy=freeway, cty=city, twn=town, ngt=night. GB=size of recording in gigabytes. Dur.=duration of recording in seconds.

ing to train more architectures using more of the data.

Fig. 3 shows our very first results, obtained from a CNN with 4 convolutional layers, each with 8 feature maps and using 3x3 kernels and trained on a single 1.5h recording. Each layer is followed by 2x2 max pooling. The final feature map layer is mapped to a 64-unit fully connected (**FC**) layer. The FC layer is mapped to an output steering angle in the range $\pm 180°$. The DVS and APS inputs were subsampled to 80x60 images. Input frame normalization was done as in Moeys et al. (2016).

Our quantitative accuracy results are too inconclusive to report but we have verified the usability of the dataset and tools. Further analysis is necessary and this work is ongoing.

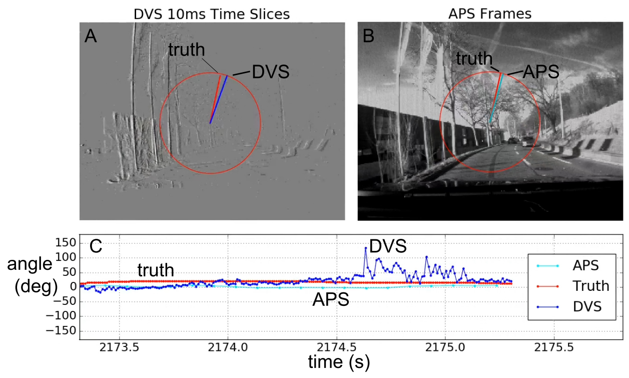

Figure 3. Steering prediction initial result. Compares our first APS and DVS steering prediction result. **A**: DVS frame and CNN output. **B**: APS frame and CNN output. **C**: segment of time history.

5. Conclusion

The main result of this paper is to introduce the DDD17 first open dataset of DAVIS driving data with end-to-end labeling, along with necessary software tools. A preliminary study on an end-to-end CNN and CNN/RNN networks show usability of the data. We are working to present applications of this data at the workshop.

Acknowledgements

We thank Dimitri Rettig and Anna Stockklauser for their help with recording the data, and inilabs and the INI Sensors Group for device support. This work was made possible by funding from the EU projects SeeBetter and Visualise .

References

Bojarski, Mariusz, Del Testa, Davide, Dworakowski, Daniel, Firner, Bernhard, Flepp, Beat, Goyal, Prasoon, Jackel, Lawrence D, Monfort, Mathew, Muller, Urs, Zhang, Jiakai, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Brandli, C., Berner, R., Yang, M., Liu, S-C., and Delbruck, T. A 240×180 130 dB 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014. ISSN 0018-9200. doi: 10.1109/JSSC.2014.2342715.

LeCun, Yann, Muller, Urs, Ben, Jan, Cosatto, Eric, and Flepp, Beat. Off-Road Obstacle Avoidance through End-to-End Learning. In *Advances in Neural Information Processing Systems*, pp. 739–746, 2005.

Lichtsteiner, P., Posch, C., and Delbruck, T. A 128x128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43 (2):566–576, Feb 2008. ISSN 0018-9200. doi: 10.1109/JSSC.2007.914337.

Lungu, Iulia-Alexandra, Corradi, Federico, and Delbruck, Tobias. Live Demonstration: Convolutional Neural Network Driven by Dynamic Vision Sensor Playing RoShamBo. In *2017 IEEE Symposium on Circuits and Systems (ISCAS 2017)*, Baltimore, MD, USA, 2017.

Moeys, D. P., Corradi, F., Kerr, E., Vance, P., Das, G., Neil, D., Kerr, D., and Delbrück, T. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pp. 1–8, June 2016. doi: 10.1109/EBCCSP.2016.7605233.

Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T. Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output. *Proceedings of the IEEE*, 102(10):1470–1484, October 2014. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2346153.

Santana, Eder and Hotz, George. Learning a driving simulator. *arXiv preprint arXiv:1608.01230*, 2016.