

# Canadian Adverse Driving Conditions Dataset

Matthew Pitropov<sup>1</sup>, Danson Garcia<sup>2</sup>, Jason Rebello<sup>3</sup>, Michael Smart<sup>4</sup>,  
 Carlos Wang<sup>4</sup>, Krzysztof Czarnecki<sup>1</sup> and Steven Waslander<sup>3</sup>

**Abstract**—The Canadian Adverse Driving Conditions (CADC) dataset was collected with the Autonomoose autonomous vehicle platform, based on a modified Lincoln MKZ. The dataset, collected during winter within the Region of Waterloo, Canada, is the first autonomous vehicle dataset that focuses on adverse driving conditions specifically. It contains 7,000 frames collected through a variety of winter weather conditions of annotated data from 8 cameras (Ximea MQ013CG-E2), Lidar (VLP-32C) and a GNSS+INS system (Novatel OEM638). The sensors are time synchronized and calibrated with the intrinsic and extrinsic calibrations included in the dataset. Lidar frame annotations that represent ground truth for 3D object detection and tracking have been provided by Scale AI.



Fig. 1. Autonomoose, our autonomous vehicle testing platform.

## I. INTRODUCTION

The safe operation of self-driving cars across a variety of adverse weather conditions remains a challenging open problem in robotics. Perception tasks become more difficult as precipitation, and in particular snowfall, degrades sensor returns and obfuscates the surroundings. Existing methods for localization and mapping, object detection, semantic segmentation, tracking and prediction, all suffer as snowfall thickens and accumulates on the ground and sensor optics, thereby changing the appearance of the driving environment.

<sup>1</sup>Department of Electrical and Computer Engineering, University of Waterloo, Canada

<sup>2</sup>Department of Electrical and Computer Engineering, University of Toronto, Canada

<sup>3</sup>Institute for Aerospace Studies, University of Toronto, Canada

<sup>4</sup>University of Waterloo, Canada

In this work, we present an autonomous driving dataset filled with representative examples of winter driving conditions in the Waterloo region, in an effort to compel the self-driving vehicle research community to address the shortcomings of current methods in adverse weather. Covering 75 driving sequences and over 20 km of driving, with varying levels of both traffic and snowfall, the Canadian Adverse Driving Conditions (CADC<sup>1</sup>) dataset includes a wide range of perception challenges collected over 3 days in the Waterloo region in Ontario, Canada, during March 2018 and February 2019. We describe the technical details of the data collected and object detection labels provided with the CADC dataset below.

## II. RELATED DATASETS

As of December 2019, there has not been a release of a multi-modal dataset for autonomous vehicles containing annotated lidar data and RGB images with the focus on adverse driving conditions and specifically snowy weather. The KITTI dataset [1] released 2012 was the first multi-modal dataset for autonomous vehicles which contained data for 4 cameras, 1 lidar and a GPS/IMU system. All cameras were forward facing and all annotated objects are within this forward facing direction. The dataset also only contains clear weather conditions.

In 2016, the Oxford RoboCar Dataset [2] was released. It contained 1 trinocular stereo camera, 3 cameras, 2 2D lidars, 1 3D lidar and 1 GPS/IMU system. Notably, the dataset was collected in several different weather conditions; however, only one drive was taken during snow. It was released with a focus on localisation and mapping and as such, no 2D or 3D object annotations are included.

In 2018, two new datasets were released, the ApolloScape Open Dataset [3] and the KAIST dataset [4]. ApolloScape contains 2 forward facing cameras, 2 lidars and 1 GPS/IMU. It contains 70,000 3D fitted car annotations. It also contains a variety of weather conditions, including bright sunlight, cloud cover and rain, and the authors suggest that they will be adding more weather conditions such as snow in the future. The Kaist Dataset contains 2 RGB cameras, 1 thermal camera, 1 lidar and 1 GPS/IMU. It includes 3D object annotations for objects visible in the forward-facing cameras, but does not include diverse weather conditions or snowfall.

More recently, the H3D Dataset [5], nuScenes Dataset [6], Argoverse Dataset [7], A\*3D Dataset [8] and the Waymo

<sup>1</sup>Pronounced cad-see. The dataset is available at <http://cadcd.uwaterloo.ca>

Open Dataset [9] have been released. H3D contains 3 cameras, 1 lidar and GPS/IMU. All cameras point in the forward-facing direction and the weather is not diverse. nuScenes contains 40,000 annotated frames, which contain 1 lidar, 6 cameras and 5 radar. Argoverse contains data from 7 cameras, 2 stereo cameras, 2 lidars as well as detailed map data. The A\*3D dataset contains 39,179 labeled point cloud frames with two front facing cameras for stereo vision. It contains data collected in the sun, cloud and rain. The Waymo Open Dataset is one of the largest publicly available autonomous driving datasets at 200,000 frames, and contains data for 1 mid-range lidar, 4 short-range lidars and 5 cameras. Both the nuScenes, Argoverse and Waymo dataset provide 360° FOV coverages with their camera configurations. They also list diverse weather conditions as part of their contributions, but currently none of them contains sequences with snow. There are therefore no comparable datasets available that capture both surround vision and lidar data as well as ground truth motion and 3D object labels in snow-filled driving conditions.

### III. VEHICLE AND SENSORS

The Autonomoose (Figure 1) is an autonomous vehicle platform created as a joint effort between Toronto Robotics and AI Lab (TRAIL)<sup>2</sup> at the University of Toronto and Waterloo Intelligent Systems Engineering Lab (WISE Lab) at the University of Waterloo. This platform has been developed to demonstrate research progress on perception, planning, control and safety assurance methods for autonomous driving. In addition, a complete autonomous driving software stack has been developed and tested on over 100 km of public road driving.

Table I provides a brief summary of the sensor outputs that are included in the CADC dataset. It contains information for our GPS/IMU, cameras, lidar, Xsens IMUs and Dataspeed Advanced Driver Assistance Systems (ADAS) Kit. The ADAS Kit provides feedback (as well as actuation) through drive-by-wire, including 4x wheel speeds, steering, throttle, brake, gear, turn signals, and other vehicle information. More detail is provided in Section Raw Data.

#### A. Time Synchronization

Each sensor data folder has a `timestamps.txt` file containing timestamps corresponding to the data files. On the Autonomoose, sensors either directly sync to the GPS time or the computer's system time. Figure 3 shows how sensor data and time synchronization are distributed across the platform. The Autonomoose Time Synchronizer is a custom-designed signal distribution board, using two ADG3123 CMOS to high voltage level translators.

<sup>2</sup>Formerly the Waterloo Autonomous Vehicles Laboratory (WAVELab) at the University of Waterloo

<sup>3</sup>Precise Point Positioning

<sup>4</sup>Real-time Kinematic

TABLE I  
SENSOR SUITE

Sensor	Parameter	Description
1x Velodyne VLP-32C Lidar	Data Rate	10 Hz
	Beam Count	32
	Range	200 m
	Field of View (FOV)	
	Horizontal	360°
	Vertical	40° (-25° to 15°)
	Angular Resolution	
	Horizontal	0.2°
	Minimum Vertical	0.3° (non-linear dist.)
	Distance Accuracy	3 cm
	Rate	~600,000 points/s
8x Ximea MQ013CG-E2 Camera	Data Rate	10 Hz
	Resolution	1280x1024 (1.3 MP)
	1x Horizontal FOV	102.8°
	8x Horizontal FOV	360°
	Sensor Size	1/1.8"
	Type	Global Shutter
	Exposure Range ( $t_{\text{exps}}$ )	800 μs to 1000 μs
	Acquisition Time	29 μs + $t_{\text{exps}}$
8x Edmund Optics C Series Fixed Focal Length Lens	Focal Length	3.5 mm
	Working Distance	0 to ∞ mm
	Aperture	f/8
1x Dataspeed Advanced Driver Assistance Systems (ADAS) Kit	Data Rate	
	Brake	50 Hz
	Gear	20 Hz
	Steer	50 Hz
	Surround	5 Hz
	Throttle	50 Hz
	Wheel Speed	100 Hz
	Miscellaneous	20 Hz
1x Novatel OEM638 Triple-Frequency GNSS Receiver	Data Rate	
	INSPVAX	20 Hz
	BESTPOS	20 Hz
	INSCOV	1 Hz
	RTK	100 Hz
	PPP <sup>3</sup> Accuracy	4 cm
	RTK <sup>4</sup> Accuracy	1 cm + 1 ppm
	Time Accuracy	20 ns RMS
	Velocity Accuracy	0.03 m/s RMS
1x Sensoor STIM300 MEMS IMU	Data Rate	100 Hz
	Bias Stability	
	Accelerometer	0.05 mg
	Gyroscope	0.5%h
	Attitude Accuracy	
	Roll/Pitch	0.015°
	Heading	0.08°
2x Xsens MTI-300-AHRS IMUs	Data Rate	200 Hz
	Bias Stability	
	Accelerometer	15 μg
	Gyroscope	10%h (MTI-300), 18%h (MTI-30)
	Attitude Accuracy	
	Average Roll/Pitch	0.3° (MTI-300), 0.5° (MTI-30)
	Average Heading	1.0°

1) *GPS Timestamps*:: From Figure 3, the Sensoor IMU is synchronized to GPS reference clock from the Novatel OEM638 receiver, which is also used as the primary reference clock on the Autonomoose computer. Two VEXXIS GNSS-502 Antennae provides the GPS radio signals to the Novatel OEM638 GNSS receiver. The receiver outputs GPS NMEA messages which contain a UTC timestamp, and a Pulse Per Second (PPS) signal set to rising edge output at the start of every second. These signals are sent to the computer, lidar and Xsens IMUs. The GPS also outputs a variable frequency (VARF) signal which is set to the rising edge output at 10 Hz (synchronized to the PPS signal). The VARF signal is used to hardware trigger the 8x cameras to start their next image acquisition cycle. Both the VARF and PPS signals are sent across the system via the I/O COM.



Fig. 2. Top: Cropped picture of Autonomoose. Bottom: Side profile and overhead view of the Autonomoose CAD file with each sensor frame axis overlaid.

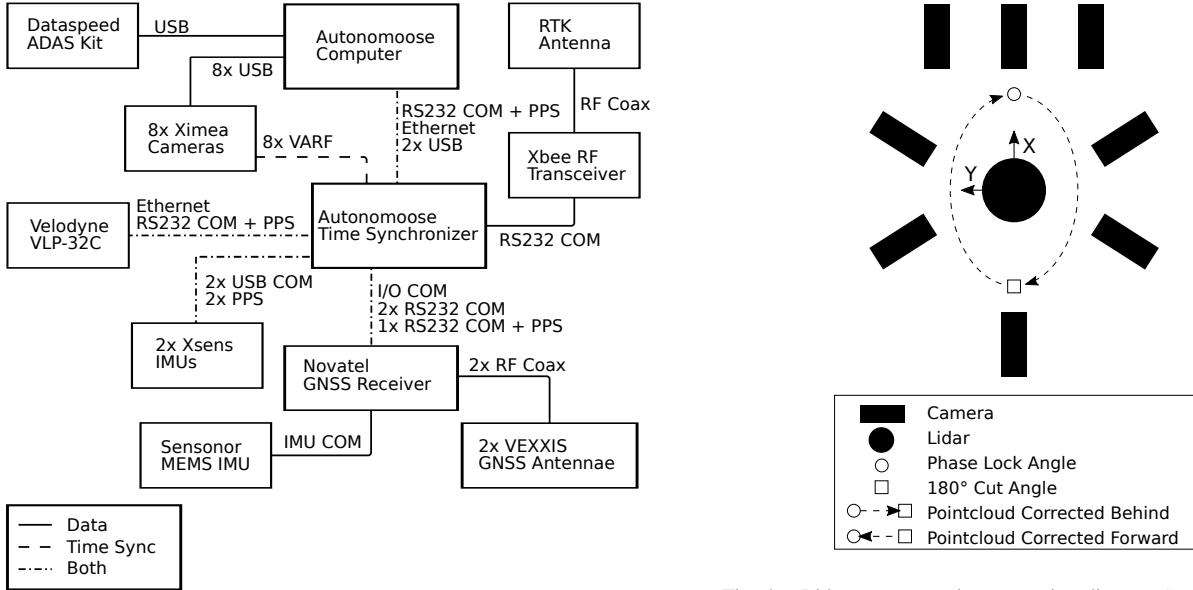


Fig. 3. Sensor communication and synchronization diagram.

2) *Computer Timestamps*:: With the GPS as the source, the Autonomoose computer receives the GPS PPS signals and NMEA messages from the COM port through a modified DB9 serial cable. Using GPSD to process the NMEA messages, as well as pps-tools to interface the Kernel PPS (KPPS) from the Linux OS, the GPS reference clock is made available in the computer system. Chrony, an implementation

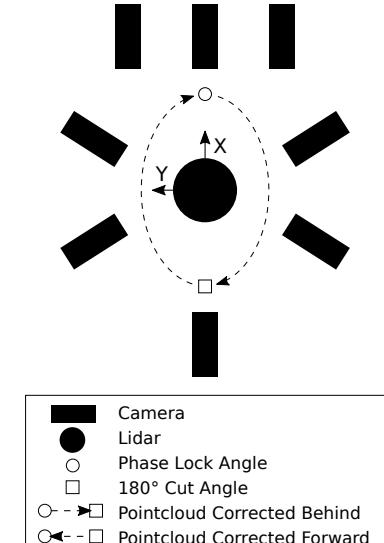


Fig. 4. Lidar-camera motion correction diagram (top-down view).

of the Network Time Protocol (NTP) is then used to synchronize the GPS reference clock with the system clock of the computer and correct for clock drift from accumulating over time. With the KPPS support the computer is stated to have a typical accuracy of  $\pm 1 \mu\text{s}$  within the GPSD Linux documentation[10].

3) *Lidar Timestamps*:: The lidar is time synchronized with the GPS PPS signal and NMEA messages. Each lidar point cloud contains a full 360° sweep of the lidar beams,

starting from the 180° cut angle (directly behind the car) and rotating clockwise, as shown on Figure 4. The message is timestamped to the phase lock angle, which is at the point when the sensor array is aligned with the positive x-axis of the lidar frame (front side of Autonomoose). The sensor array passes the phase lock angle at every rising edge from the VARF signal and motion corrected messages are corrected to this timestamp. During motion correction, the lidar points on the left hemisphere are transformed forward in time, while the right hemisphere are transformed backward in time based on the vehicle motion to match the time at the phase lock angle. Through this process, the point cloud looks stationary and is synchronized to the VARF signal.

4) *Cameras Timestamps*:: As shown on Table I and Figure 3, all 8 cameras are hardware triggered at 10 Hz with the VARF signal from the GPS. Camera images are taken at the same instant when the lidar passes the phase lock angle. Unfortunately, the Ximea MQ013CG-E2 does not have an internal time and therefore can not directly timestamp its own data. Therefore it is timestamped when the image is acquired in Robot Operating System (ROS), and is also truncated to within 0.1 s to match the VARF output rate. This truncation is based on the validity that under normal operation the image acquisition period is well below the 0.1 s VARF period (see Table I acquisition delay). By doing this the image timestamp should more accurately represent the time which the image is captured by the camera.

5) *Vehicle Control Timestamps*:: The primary vehicle control messages required for an automated driving system (ADS) are the vehicle's throttle, brake, steering and gear shift. These control messages, received from the vehicle's controller area network (CAN) bus, are decoded by the Dataspeed ADAS kit and are sent to the computer via a Universal Serial Bus (USB) port. Since the Dataspeed ADAS kit cannot be directly connected to a GPS PPS signal or an NMEA message, each decoded message is timestamped with the latest computer timestamp once the ROS CAN driver receives it. All CAN Bus messages are sent to the CAN-bus-to-USB converter, which is part of the Dataspeed ADAS Kit that relays them to the Autonomoose computer via USB, as soon as possible. There are two exceptions. One is the surround report which is event periodic and depends on potential alerts. Another is the miscellaneous message which is aggregated over the cycle time. Table II contains the standard deviation of the period for each Dataspeed decoded message on one drive showcasing low standard deviations for each message.

6) *Xsens Timestamps*:: Time synchronization for the Xsens IMUs is a two step process that involves receiving PPS from the GPS and timestamps from the Computer. As these devices cannot receive NMEA messages directly, any error in the computer time will propagate to these IMUs.

#### IV. SENSOR FRAMES

Figure 2 contains the full sensor diagram for Autonomoose. The origin of the base\_link frame is at the center of the rear axle with x pointing forwards, y to the left and

TABLE II  
THE STANDARD DEVIATION OF THE PERIOD FOR EACH DATASPEED MESSAGE OVER DRIVE 0027 ON 2019.02.27.

Message Type	Standard deviation (ms)
brake.info_report	1.849
brake_report	1.036
gear_report	0.743
misc_1_report	1.162
steering_report	1.054
surround_report	0.671
throttle_report	1.042
wheel.speed.report	2.431

z upwards. The origin of the GPS/IMU frame is within the trunk inside the IMU. It is rotated 90° clockwise along the z axis compared to the base\_link frame. The lidar frame has the same orientation as the base\_link and is located at the lidar's optical center. Each camera frame is located at its respective optical center, with z axis pointing towards the image plane, x axis to the right and y axis downwards. The three forward-facing cameras are aligned in a trinocular configuration, allowing for convenient rectification if needed. Each Xsens IMU is placed on top of the camera with the same orientation of the base\_link but with its x axis being aligned with the camera's z axis. Small misalignments exist in each sensor axis position and exact transforms are included with the extrinsic calibration file.

#### V. CALIBRATION

Figure 5 details the contents of a calibration zip file which is discussed further in this section.

```
date.calib.zip
└── 0X.yaml X={0,1,2,3,4,5,6,7}
    └── extrinsics.yaml
```

Fig. 5. Contents of a calibration zip file, which contains intrinsics for each 8 cameras, and extrinsics between each pair of sensors.

##### A. Camera intrinsics

Each of the eight camera yaml files labeled 00.yaml to 07.yaml store intrinsic calibration information: camera name, height and width (pixels), camera matrix, distortion model and the distortion coefficients.

Equation 1 contains the camera matrix  $K$ . The focal length is  $f_x$  and  $f_y$  in pixels, the axis skew  $s$  causes shear distortion and the coordinates for the optical center are  $c_x$  and  $c_y$  in pixels.

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

We provide the distortion coefficients for the Brown-Conrady or “Plumb Bob” distortion model [11]. The ordering of coefficients in equation 2 is also used in OpenCV, Matlab and ROS. The values for  $k_1$ ,  $k_2$ ,  $k_3$  are the 2nd, 4th and 6th order radial coefficients, respectively. The two tangential distortion coefficients are  $p_1$  and  $p_2$ .

$$k_c = [k_1 \ k_2 \ p_1 \ p_2 \ k_3] \quad (2)$$

## B. Sensor extrinsics

The `extrinsics.yaml` file contains an array of  $4 \times 4$  homogenous transformation matrices. There are transforms between these pairs of frames: base\_link to lidar, GPS to lidar, camera to right neighbouring camera and camera to lidar. Each transform has been attained through a calibration procedure with two exceptions. The lidar to base\_link transform was measured using a 3D scan of Autonomoose and direct camera to lidar transforms for each camera is provided for convenience. Equation 3 contains an example transform converting data within the GPS/IMU frame ( $\mathcal{F}_G$ ) to the lidar frame ( $\mathcal{F}_L$ ). The rotation matrix from  $\mathcal{F}_G \rightarrow \mathcal{F}_L$  is defined as  $R_{LG} \in \mathbb{R}^{3 \times 3}$ . The translation matrix from  $\mathcal{F}_G \rightarrow \mathcal{F}_L$  is defined as  $t_{LG} \in \mathbb{R}^{3 \times 1}$ .

$$T_{LG} = \begin{bmatrix} R_{LG} & t_{LG} \\ 0 & 1 \end{bmatrix} \quad (3)$$

## VI. DATASET FORMAT

The CADC dataset has been stored in a format similar to the KITTI Raw dataset. Sensor data are provided for each recorded drive, along with the calibration data for each given calendar day of recording. Each drive has raw output from all sensors at their full sample rate; labeled data, which is a sampled subset of the raw dataset at  $\sim 3$  Hz; and lastly the corresponding 3D annotations for the labeled data. Figure 6 contains the file structure used for all data available in this dataset.

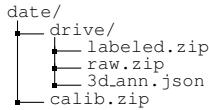


Fig. 6. Folder structure for downloading CADC.

There are three types of sensor folders, which are all displayed in Figure 7. Camera image folders contain PNG images. Lidar point clouds are stored as binary files, which can be read using the development kit. All other sensor data have text files with space-separated values. Detailed information for each value can be found within the sensor's `dataformat.txt` file.

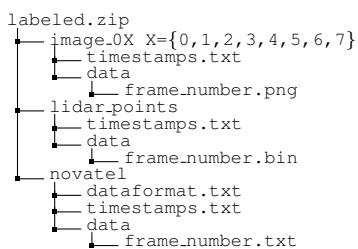


Fig. 7. Contents of a labeled drive zip file, which contains data from 8 cameras, 1 lidar and 1 GPS/IMU.

## VII. RAW DATA

A raw data zip file contains content from all available sensors on Autonomoose. Figure 8 contains the complete list.

There is a subsection for each sensor: camera images, lidar, novatel GPS/IMU, vehicle control and the Xsens IMUs. The data rate for each message is included within Table I.

```

raw.zip
  ├── image_0X X={0,1,2,3,4,5,6,7}
  ├── lidar.points
  ├── lidar.points.corrected
  ├── novatel
  ├── novatel.bestpos
  ├── novatel imu
  ├── novatel inscov
  ├── novatel rtk
  ├── brake.info.report
  ├── brake.report
  ├── gear.report
  ├── misc1.report
  ├── steering.report
  ├── surround.report
  ├── throttle.report
  ├── turn.signal.cmd
  ├── wheel.speed.report
  ├── xsens.300 imu
  ├── xsens.300 velocity
  ├── xsens.30 imu
  └── xsens.30 velocity
  
```

Fig. 8. Contents of a raw data zip file.

### A. Image data

There are 8 camera image folders, one for each of the 8 cameras denoted `image_0X` for X from 0 to 7. Driving sequence 0066 from 2019\_02\_27 contains 15 missing `cam_00` images, this segment was not chosen to be part of the labeled data.

1) `images::`: Each original image has been stored as a 1280x1024 PNG image with a size of  $\sim 2$  MB.

### B. Lidar data

1) `lidar.points::`: Each lidar point cloud file has a timestamp which corresponds to when the lidar beams pass the phase lock angle.

2) `lidar.points.corrected::`: Each lidar point cloud file has been corrected for motion distortion relative to vehicle motion using post-processed GPS/IMU data to the timestamp which corresponds to when the lidar beams pass the phase lock angle.

### C. GPS/IMU data

1) `novatel::`: A Novatel Inertial Position Velocity Attitude - Extended (INSPVAX) message, which contains the most recent position, velocity and orientation as well as standard deviations.

2) `novatel_bestpos::`: A Novatel Best Position (BESTPOS) message, which contains the best available current position.

3) `novatel_imu::`: IMU message, which contains the change in orientation and acceleration about each axis corrected for gravity, the earth's rotation and sensor errors. Pitch, roll and yaw are defined as right handed with pitch about the x axis, roll about the y axis and yaw about the z axis. To retrieve instantaneous acceleration or rotation, the data should be multiplied by the sample rate of 100 Hz.

4) `novatel_inscov::`: A Novatel Inertial Covariance (INSCOV) message, which contains the uncertainty of the current pose solution. These are three  $3 \times 3$  matrices, each with the variance along the diagonal corresponding to position, attitude and velocity.

5) *novatel\_rtk*:: A Novatel RTK message, created by post-processing the GPS data with base station data. It contains a subset of the INSPVAX message: the position and orientation as well as standard deviations.

#### D. Vehicle control data

Each data type was converted from their respective Dataspeed ROS message, with detailed information provided within their *dataformat.txt* file as well as online at ROS dbw\_mkz\_msgs. The messages contain information on braking, gear, steering, surround, throttle and wheel speed, sorted by folder. The miscellaneous report messages contain data from various sensors, for example the state of seat belts, doors and the outside air temperature.

#### E. Xsens IMU data

1) *xsens\_300\_imu* and *xsens\_30\_imu*:: Contains the orientation (with respect to the ENU frame), angular velocity ( $\text{rad s}^{-1}$ ) and linear acceleration ( $\text{m s}^{-2}$ ) as well as the corresponding covariance matrices.

2) *xsens\_300\_velocity* and *xsens\_30\_velocity*:: Contains the linear velocity ( $\text{m s}^{-1}$ ) and angular velocity ( $\text{rad s}^{-1}$ ).

## VIII. LABELED DATA

Object labels are provided for a subset of the raw data. The subset is obtained by down sampling the raw data to select one frame every 300 ms for a total of 50-100 frames per drive (15-30 seconds in length). As shown in Figure 7, there are 10 folders, one for each of the eight cameras denoted *image\_0X* for X from 0 to 7, a *lidar\_points* folder for the Velodyne lidar data and a *novatel* folder containing GPS/IMU data. All data is synced such that data for all sensors was matched to the closest data timestamp of the GPS/IMU.

#### A. Data ( $\sim 3 \text{ Hz}$ )

1) *images*:: This data is equivalent to the image data in the Raw Data section with the only difference being that these images have been undistorted.

2) *lidar\_points*:: These lidar points are equivalent to the motion corrected lidar points in the Raw Data Section.

3) *novatel (GPS/IMU)*:: This data is equivalent to the post processed Novatel RTK data within the Raw Data section.

## IX. 3D ANNOTATIONS

Each 3D annotation file contains a list of frames with the detected cuboids in each frame. Figure 9 contains the structure of this JSON file with an example cuboid.

#### A. cuboid

1) *uuid*:: A string used to identify this cuboid across all frames within a drive.

2) *camera\_used*:: The index of the camera used to label this cuboid.

3) *position*:: The center of the object in the lidar frame. The object has its z axis pointing upwards, x axis pointing to the forward facing direction of the object and the y axis pointing to the left direction of the object.

```
[
  {
    "cuboids": [
      {
        "uuid": "0241bd75-b41f-4c67-8fcf-9388fd7e2c8b",
        "label": "Pedestrian",
        "position": {
          "x": -48.85306519173028,
          "y": -10.954928897518318,
          "z": -0.728937152344576
        },
        "dimensions": {
          "x": 0.828,
          "y": 0.766,
          "z": 1.688
        },
        "yaw": -0.03539451751530657,
        "stationary": false,
        "camera_used": 7,
        "attributes": {
          "age": "Adult"
        },
        "points_count": 11
      },
      ...
    ],
    ...
  ]
]
```

Fig. 9. The JSON structure of a 3D annotation file.

TABLE III  
UNIQUE ATTRIBUTES.

Relevant Label(s)	Attribute	Values
Truck	truck_type	Snowplow.Truck Semi.Truck Construction.Truck Garbage.Truck Pickup.Truck Emergency.Truck
Bus	bus.type	Coach.Bus Transit.Bus Standard.School.Bus Van.School.Bus
Bicycle	rider.state	With_Rider Without_Rider
Pedestrian Pedestrian_With_Object	age	Adult Child
Traffic_Guidance_Objects	traffic_guidance_type	Permanent Moveable

4) *dimensions*:: The dimensions of the cuboid with x being the width from left to right, y being the length from front to back and z being the height from top to bottom.

5) *yaw*:: The orientation of the object in the lidar frame. A yaw of zero will occur when the cuboid is aligned with the positive direction of the lidar frame's x-axis. A yaw value of  $\pi/2$  occurs when the cuboid is aligned with the positive direction of the lidar frame's y-axis.

6) *stationary*:: A boolean value that describes if the object is stationary across all frames in a drive.

7) *points\_count*:: An integer value of how many lidar points are contained within the cuboid at this frame.

8) *label and attributes*:: Each object has a label as well as any attributes that are defined for the specified label. The following labels are used: Car, Truck, Bus, Bicycle, Horse and Buggy, Pedestrian, Pedestrian with Object, Animal, Garbage Container on Wheels, and Traffic Guidance Object.



Fig. 10. A map of data collected for CADC.

TABLE IV

TABLE CONTAINING DRIVING ASPECTS OF CADC DATASET.

Dataset type	# of point clouds	# of images	Distance (Km)
Raw	32887	263637	20.33
Labeled	7000	56000	12.94

TABLE V

TABLE CONTAINING DATA ASPECTS OF CADC DATASET.

Dataset type	Compressed size (GB)	Uncompressed size (GB)
Raw	472.7	514.73
Labeled	92.76	97.79

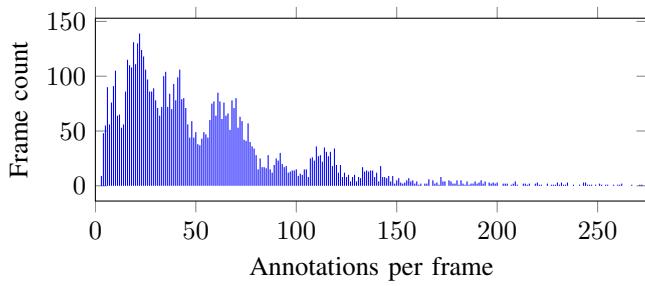


Fig. 11. Each frame has been binned by the number of cuboid annotated.

The set of vehicle classes includes [Car, Truck, Bus, Bicycle, Horse and Buggy] and have attributes [Parked, Stopped, Moving] in every frame. A Horse and Buggy label is included, although there is only one in the training data. Pedestrian, Pedestrian with Object, Animal, Garbage Container on Wheels and Traffic Guidance Object are not in vehicle class.

Table III lists potential attributes that are unique based

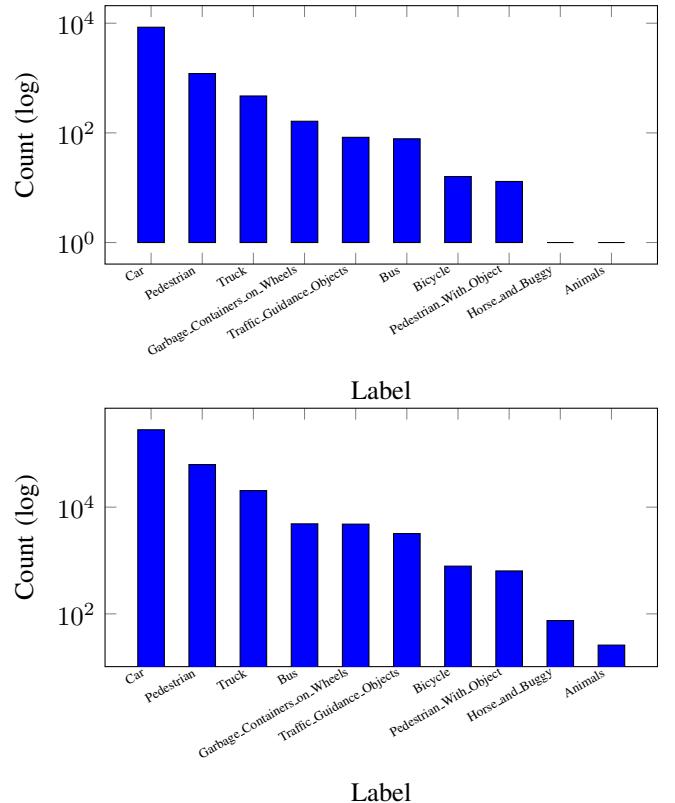


Fig. 12. Top: Number of unique labeled objects across all frames. Bottom: Number of object instances across all frames.

on the given object class. The Truck label's major attributes are pickup trucks and semi-trucks. The Bus label has four different attributes: two transit types for coach and public transit and two school types for standard and van-length school buses. The Bicycle label has an attribute for whether it is, or is not, being ridden. The Pedestrian label has an age attribute set to Adult or Child. A Pedestrian with Object also exists and is used, for example, when a person is pushing a stroller. The Animal label is used for wild animals as well as pets. The Garbage Container on Wheels label is used for tall garbage containers. Lastly, Traffic Guidance Objects could be pylons with the Moveable attribute or permanent vertical delineators for bike lanes with the Permanent attribute.

The Car label contains cars, SUVs and vans, and has no unique attributes. One drawback of this design is that the single label covers several vehicle body sizes. We plan to subsequently release 2D annotations that introduce attributes to subdivide this class.

## X. DATASET STATISTICS

We provide several figures and tables showcasing the wide range of data that has been collected. Figure 10 shows a map with GPS points placed from the data. The red points represent the full dataset, whereas the overlayed green points represent the subset of the data that is labeled.

Table IV contains information on the number of point clouds, images and distance traveled. Table V contains the

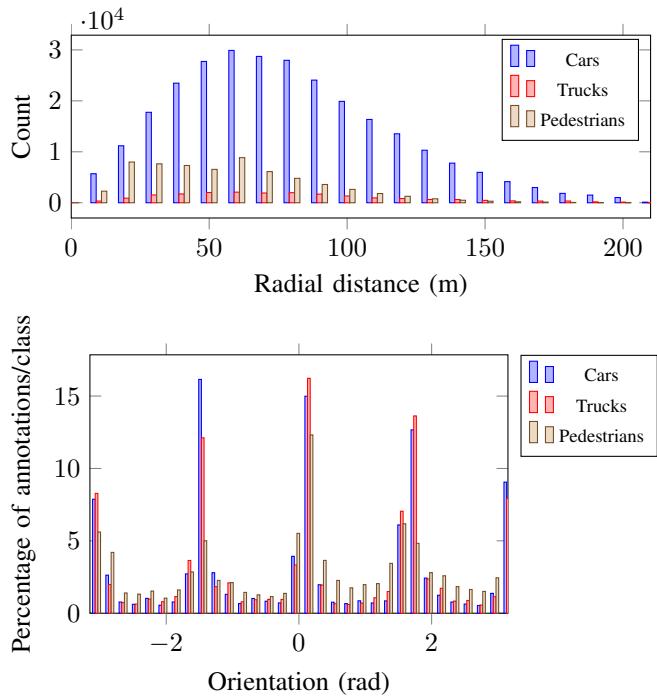


Fig. 13. Top: Radial distance from the center of a cuboid to the origin of the lidar frame. Bottom: Orientation of cuboids within the lidar frame.

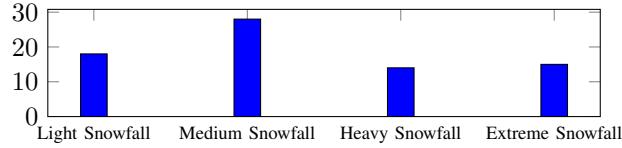


Fig. 14. Number of drives for each level of snowfall.

compressed and uncompressed sizes of the raw and labeled data. They are similar due to the images being the largest sensor data size and already losslessly compressed in the raw data.

Figure 11 shows the number of frames for each number of annotations per frame. This is restricted to objects with at least one lidar point within its cuboid for a specific frame. Figure 12 contains two graphs. The top graph shows the number of unique objects binned by their label across all frames based on uid. The bottom graph displays the number of object instances by how many frames they appear in. Figure 13 also contains two graphs for the three objects with the most instances. The first displays the radial distance of objects, and the second displays the orientation of each object. Table VI contains the number of instances for each vehicle label and type. Table VII contains the number of instances for each pedestrian with age attribute.

Each drive has been given a snowfall level from: light, medium, heavy and extreme. This was done by first cropping the point cloud to a cube that spans -4 to 4 in the x and y direction and -3 to 10 in the z direction within the lidar frame, and then applying the Dynamic Radius Outlier Removal (DROR)[12], which is designed to remove snowfall from

TABLE VI  
TOTAL INSTANCES FOR EACH VEHICLE LABEL SORTED BY STATE ATTRIBUTE.

Label (and attribute)	Total	Parked	Stopped	Moving
Car	281941	193246	18002	70693
Truck	20411	9959	2060	8392
Snowplow_Truck	1497	628	221	648
Semi_Truck	4653	1547	516	2590
Construction_Truck	715	433	45	237
Garbage_Truck	27	1	0	26
Pickup_Truck	13045	7178	1138	4729
Emergency_Truck	474	172	140	162
Bus	4867	476	1513	2878
Coach_Bus	751	193	135	423
Transit_Bus	2899	196	847	1856
Standard_School_Bus	983	0	531	452
Van_School_Bus	234	87	0	147
Bicycle	785	520	0	265
With_Rider	265	0	0	265
Without_Rider	520	520	0	0
Horse_and_Buggy	75	0	75	0

TABLE VII  
TOTAL INSTANCES FOR EACH PEDESTRIAN LABEL SORTED BY AGE ATTRIBUTE.

Label	Total	Adult	Child
Pedestrian	62851	61664	1187
Pedestrian_With_Object	638	638	0

LIDAR point cloud data. The number of points removed is taken to be the approximate number of snow reflectance points in a LIDAR scan. Bins for snowfall intensity, depicted in Figure 14 are defined as: Light (25-249), Medium (250-499), Heavy (500-749) and Extreme (750-1500). Lastly Figure 15 contains an example LIDAR scan and image for each snowfall level. The snow covering of the road is also included. There are 18 driving sequences with bare road and 57 with snow covering the road.

## XI. DEVELOPMENT KIT

A basic development kit implemented in python is available at [cadc\\_devkit](#) with the ability to view the vehicle path, the lidar projected onto the images and lastly the 3D annotations projected onto the images and lidar.

### A. run\_demo\_vehicle\_path.py

This script loads all GPS messages in a drive, converts them to an ENU frame with the origin at the first message and plots each message as an axis frame. Figure 16 is an image displaying the output of this script.

### B. run\_demo\_lidar.py

This script loads a camera image and the corresponding lidar file in a drive, loads the calibration data, then projects each lidar point onto the camera image. Point color is scaled by depth. Figure 17 is an image displaying the output of this script.



Fig. 15. Top down lidar view of each snowfall levels with the corresponding front camera image. Top: left image couple is the light snow and the right side is medium snow. Bottom: left image couple is heavy snow and the right is extreme snow.

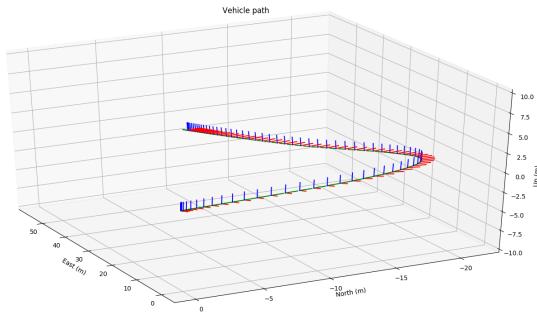


Fig. 16. Output from the `run_demo_vehicle.path.py` script on the drive 0027 from 2019\_02\_27. It shows the path of Autonomoose as it makes a left hand turn



Fig. 17. Output from the `run_demo_lidar.py` script on drive 0027 from 2019\_02\_27. The lidar points are projected onto the image of the front camera.

### C. `run_demo_tracklets.py`

This script loads a camera image and the corresponding 3D annotation file in a drive, loads the calibration data, then creates and projects each cuboid within the frame onto the camera image. Figure 18 is an image displaying the output of this script.

### D. `run_demo_lidar.bev.py`

This script loads lidar data and the corresponding 3D annotation file in a drive, then creates a birds eye view of the lidar point cloud with the cuboid boxes overlaid. Figure 19 is an image displaying the output of this script.

## XII. CONCLUSIONS

We present the CADC dataset, an annotated object dataset containing lidar and images collected within the Region of Waterloo during winter driving conditions. This dataset will enable researchers to test their object detection and localization and mapping techniques on challenging winter weather. In the future, we plan to release 2D annotations for each individual image containing truncation and occlusion values. We also will be creating a benchmark for 3D object detection using this dataset.

## ACKNOWLEDGEMENT

CADC Dataset would not have been created without the help of many members of the Autonomoose project over the past several years. Juichung Kuo, Ji-won Park and Archie Lee contributed to the initial development and setup of the sensor platform and triggering, as well as the time synchronization required for data capture. Kevin Lee supported this research through driving and collecting data with the Autonomoose. Our SLAM team, consisting of Nav Ganti, Leo Koppel and Ben Skikos created the ROS packages we used to localize Autonomoose and motion correct the lidar scans. Asha Asvathaman created several scripts for the development kit used to visualize and process the dataset. Lastly Jessen Liang designed and produced a calibration rig to make calibration more efficient. We would also like to thank Scale AI for annotating our dataset as well as the members of TRAIL and WISELab for assisting with auditing the annotations.

## REFERENCES

- [1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Robotics Research (IJRR)*, 2013.



Fig. 18. Output from the `run_demo_tracklets.py` script on drive 0033 from 2019\_02\_27. Each cuboid from the 3D annotation file is projected onto the image of the front camera.

- [2] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 Year, 1000km: The Oxford RobotCar Dataset,” *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: <http://dx.doi.org/10.1177/0278364916679498>
- [3] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, “The ApolloScape open dataset for autonomous driving and its application,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 11, 2019. [Online]. Available: <http://dx.doi.org/10.1109/TPAMI.2019.2926463>
- [4] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, “KAIST multi-spectral day/night data set for autonomous and assisted driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, March 2018.
- [5] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, “The H3D dataset for full-surround 3D multi-object detection and tracking in crowded urban scenes,” 2019.
- [6] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Lioung, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuScenes: A multimodal dataset for autonomous driving,” 2019.
- [7] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, “Argoverse: 3D tracking and forecasting with rich maps,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [8] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin, “A\*3D dataset: Towards autonomous driving in challenging environments,” 2019.
- [9] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, “Scalability in perception for autonomous driving: Waymo open dataset,” 2019.
- [10] G. E. Miller and E. S. Raymond, “Gpsd time service howto,” Jun 2018. [Online]. Available: <https://gpsd.gitlab.io/gpsd/gpsd-time-service-howto.html>
- [11] D. C. Brown, “Decentering distortion of lenses,” *Photometric Engineering*, vol. 32, no. 3, pp. 444–462, 1966.
- [12] N. Charron, S. Phillips, and S. L. Waslander, “De-noising of lidar point clouds corrupted by snowfall,” in *2018 15th Conference on Computer and Robot Vision (CRV)*, May 2018, pp. 254–261.

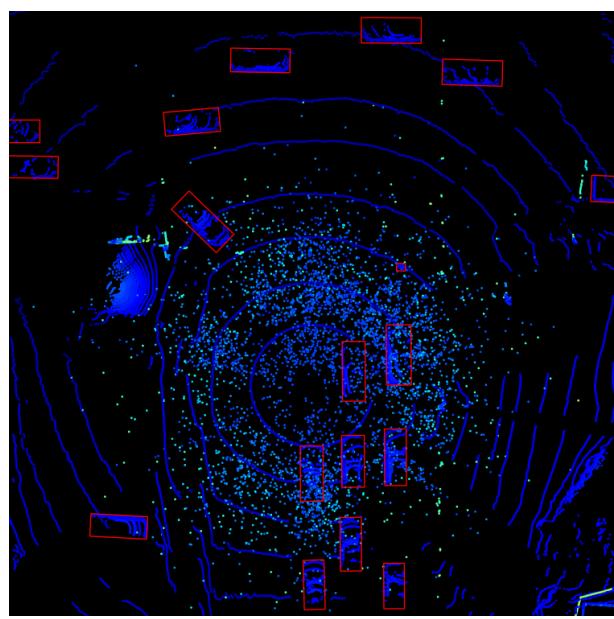


Fig. 19. Output from the `run_demo.lidar.bev.py` script on drive 0033 from 2019.02.27. Each cuboid from the 3D annotation file is displayed within the lidar frame from a top down perspective.