ISSN: 2994-6433

Research Article

Journal of Sensor Networks and Data Communications

The KI-ASIC Dataset

Daniel Scharf, Seifeddine Saadani, Alfred Hoess*

OTH Amberg-Weiden, 92224 Amberg, Germany

***Corresponding Author**
Alfred Hoess, OTH Amberg-Weiden, 92224 Amberg, Germany.

Submitted: 2023, Aug 17; **Accepted**:2023, Sep 18: **Published**: 2023, Sep 22

Citation: Scharf, D., Saadani, S., Hoess, A. (2023). The KI-ASIC Dataset. *J Sen Net Data Comm, 3*(1), 76-80.

Abstract

We present a novel dataset captured from a BMW X5 test carrier within the German research project KI-ASIC for use in radar sensor development and autonomous driving research. Our work aims at providing a blueprint for the process of creating labeled datasets for the development of neural networks for pattern recognition in radar data in the automotive environment. With a variety of different sensor types such as wide angle color cameras, a high-resolution color stereo camera, an Ouster OS1-64 laser scanner and three novel Infineon radar sensors, we recorded over 100,000 scenes of real traffic scenarios as well as defined test scenarios with a frequency of 10 Hz. The scenarios in real traffic contain inner-city situations, but also scenes from rural areas with static and dynamic objects. Besides, the defined test scenarios are based on the NCAP scenarios and focus mostly on turning, overtaking and follow-up maneuvers. The data from the different sensors is calibrated, synchronized and timestamped including raw and rectified information. Our dataset also contains labels for all detected objects from a defined class list with distance and angle properties. The content of the paper aims at the description of the recording test carrier, the format of the provided sensor data and the structure of the overall dataset.

Keywords: Data Labeling, Deep Learning, Computer Vision, Autonomous Driving, Neural Network, Radar Sensor, Lidar Sensor, Stereo Camera, Object Detection, KI-ASIC.

1. Introduction

The KI-ASIC dataset has been recorded from a moving test carrier while driving in and around Amberg, Germany. The defined test scenarios were performed at the airfield in Schmidgaden near Amberg. It includes images, laser scans and raw data of three novel Infineon radar sensors. The goal of the dataset is to provide a data basis for the development of neural networks for radar pattern recognition, but also to expand the currently available data field in literature in order to advance automated and autonomous driving [1-4]. While our introductive paper describes the process of data acquisition and semi-automated labeling using an exemplary dataset, we here close the frame by giving insight into the technical details of the used sensors and the raw data itself [5]. All collected information was then stored in a separate dataset for every accomplished test scenario. A subset of the KI-ASIC dataset can be downloaded from abc.def.ghi.jkl.mno.oth-aw.de, all data is available on request. To give a detailed overview about the labeling process that leads to the provided datasets, we refer the reader to [5,6].

2. Sensor Setup

Our setup shown in Fig. 1 includes the following sensors:

- 2x IDS UI-3080CP Rev. 2 color cameras (used as one stereo camera), 5.04 Megapixels, 2/3” Sony IMX250 CMOS, global shutter.
- 2x Kowa LM5JC10M lenses, 5 mm, 2/3” C-Mount, 10 Megapixels, manual focus with lock, opening angle horizontal 82.2◦, opening angle vertical 66.5◦.
- 2x IDS UI-3040CP Rev. 2 color cameras, 1.57 Megapixels, 1/3” Sony IMX273 CMOS, global shutter.
- 2x Theia MY125M lenses, 1.3 mm, 1/3” C-Mount, 5 Megapixels, distortion < 3%, opening angle horizontal 125◦, opening angle vertical 109◦.
- Ouster OS1-64 rotating 3D laser scanner with gradient beam distribution, 0.01◦ angular resolution and 0.1 cm range resolution, field of view: 360◦ horizontal, 45◦ vertical, range up to 200 m. The sensor was used with following configurable settings: 10 Hz, 1024x64 channels, collecting more than 2.6 million points/second.
- 3x Infineon radars, 76 GHz ± 607.7 MHz for mid-range, opening angle 180◦, 4 transmitters, 8 receivers, 64 ramps.

Figure 1: Test Carrier with Sensor Setup

Our BMW X5 test carrier is equipped with four color cameras in the area of the interior mirror, a rotating 3D laser scanner mounted on the vehicle roof and three novel radar sensors integrated into the front bumper.

The focus of the project aims at detecting objects in the field of view of our stereo camera which covers a region of about ±40◦, where most of the objects are aging. Therefore, only the stereo camera and laser scanner are calibrated to each other, the outer wide-angle cameras are turned outwards and observe the remaining opening angle of the radar sensor without being calibrated or labeled in order to just not lose any edge objects.

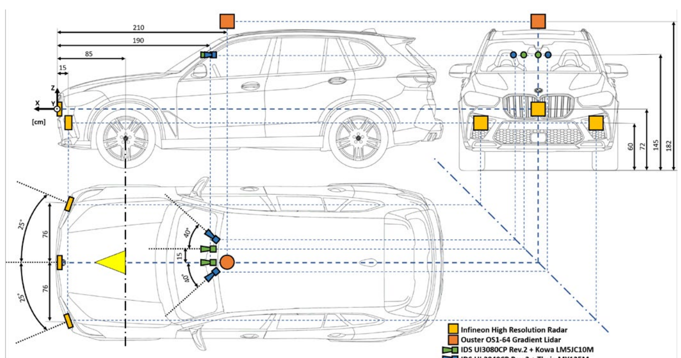

Figure 2: Sensor Setup with Mounting Positions and Coordinate System

The baseline of the stereo camera system illustrated in Fig. 2 is about 10 cm for the test drives in January 2023, but was enlarged to 15 cm for the tests in March and April 2023 to optimize the distance determination on the basis of disparity. The trunk of our test carrier houses a PC for data recording with an Intel Xeon Gold 5119T processor, 64 GB RAM, two SSDs with 8 TB each and a NVIDIA Quadro P2000 graphic card with 5 GB RAM. Our computer runs on Ubuntu Linux (64 bit) to store the incoming data streams for all sensors.

3. Dataset

We have automatically processed all recorded test scenarios with our processing pipeline, a small subset was also manually relabeled with our implemented labeling tool. The processed datasets can be accessed at abc.def.ghi.jkl.mno.oth-aw.de. The object detection was performed using two neural networks (Mask-RCNN and DetectoRS trained using the Cityscapes dataset [7,8]. The selected dataset includes significantly more object classes than typically encountered in intersection traffic. Therefore, it was decided to reduce the scope of detectable object classes to a level sufficient for the real-world traffic radar use case that is the focus of the project. Only the following object classes are considered: person, car, bicycle, rider, motorcycle, truck and bus. The various Euro NCAP scenarios are used as a basis for data acquisition with the test vehicle [9]. However, the focus of the tests is not on collision avoidance by the vehicle, but on detecting critical situations. The test scenarios are therefore set up in such a way that a collision between the test vehicle and other road users is avoided. Special attention will be paid to scenarios such as turning and tailgating maneuvers with road users typical for urban intersections in real traffic. Those tests were performed at the airfield in Schmidgaden mostly without having interfering objects. In addition, some scenarios in real traffic in and around Amberg were recorded and evaluated.

The test drives were conducted from 23rd to 27th of January, 14th of March and 6th to 12th of April 2023 with all possible weather conditions. We have processed over 100,000 scenes with a total data size of around 5 TB. The quality of the datasets was evaluated using 1,010 randomly selected scenes. Thereby, 414 objects were detected correctly, 4 incorrectly and 21 not at all in the relevant range of up to 40 meters.

3.1. Dataset Description

The dataset is divided according to test days, each of which includes both defined test and evaluated driving scenarios in real traffic. Examplary scenes are shown in Figure 3.

Figure 3: Examples from the KI-ASIC Dataset

A scenario typically comprises approx. 10 seconds in a separate folder. Each evaluated scenario comprises another folder in which all labeled objects with calculated distances are stored as an image for each timestamp.

If it was possible to determine a distance value for an object using both the lidar data and the information from the stereo camera system, two images are stored for each timestamp. In addition, the result files are saved in HDF5 format in each scenario folder [10]. Due to file size limitations, 50 timestamps including raw data and computed object information are stored in a result file whose directory structure is shown in Fig. 4. In the following, the structure of the dataset as well as the annotations for the detected objects will be described on the basis of an exemplary scene:

- **Specifications:** This includes general information such as the date of the test, the neural networks used for image labeling, the maximum distance for object detection, the name of the test scenario defined in the test catalogue, as well as the weather data at the time and the GPS coordinates of the test drives. In addition, the specifications of the sensors used to record the data necessary are also documented.
- **Images:** Each timestamp includes the image name, raw images from the wide-angle cameras, rectified and undistorted images from the stereo camera, the resulting depth map and the labeled images including the distance for each object calculated using the lidar sensor or the stereo camera.
- **Lidar Data:** The raw data of the lidar sensor were recorded in pcd format and stored with the same structure.
- **Radar Data:** The raw data of each of the three radar sensors are saved with data type int16 in a separate folder.

```
dataset.h5
├── 1_user_agreement
├── specifications
│   ├── general
│   │   ├── classes_to_be_analyzed
│   │   ├── label
│   │   ├── labeling_nn
│   │   ├── latitude
│   │   ├── longitude
│   │   ├── max_dist
│   │   ├── timestamp
│   │   ├── translation_camera_radar
│   │   └── weather_data
│   └── list_of_parameters
│       ├── cameras
│       ├── lidars
│       └── radars
├── timestamp_0001
│   ├── depth_map
│   ├── image_labeled
│   ├── image_labeled_depth
│   ├── image_left_120
│   ├── image_left_stereo
│   ├── image_name
│   ├── image_right_120
│   ├── image_right_stereo
│   ├── objects
│   ├── point_cloud
│   └── radar_data
├── timestamp_0002
│   └── ...
└── ...
```

Figure 4: Dataset Structure in HDF5 Format

3.2. Annotations

For each detected object within the field of view of the camera system, a distance in meters (mean and standard deviation) as well as a horizontal and vertical angle range in rad (mean and value to outer edge) is calculated under which the test object occurs to the front-mid radar sensor. When possible, this calculation was performed using both the stereo camera system and the lidar data if enough lidar points lay within the object mask. Every detected object with a maximum distance of 40 meters was considered, objects further away or without determinable distance were filtered out. The left camera of the stereo camera system is used as the origin of the calculations. The outcomes of the property computation were then transformed to the front-mid radar sensor using the measured translation between the camera and radar. Fig. 5 shows the result of an exemplary test scenario.

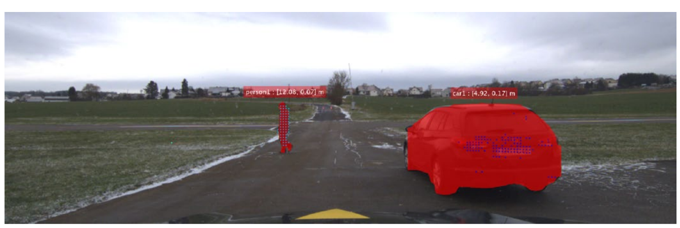

Figure 5: Exemplary Test Scenario with Detected Objects

The processing chain in MATLAB has detected two objects within the scene of the defined test scenario, whose object classes, contours as well as distance information starting from the front-mid radar sensor have been inserted into the image.

In addition, a 2D bounding box in pixel starting from the upper left corner is specified, which indicates in which image area in pixels the detected object is located. For each detected object, the object mask is saved as a bitmap with the same resolution as the original image and a confidence score is given, which indicates the probability of belonging to the particular object class. Comparing the maximum number of lidar points that can ideally occur within the object mask with the lidar points actually present within the object mask after merging camera and lidar data, a quality score can be determined. For a detailed description of the object property determination, the reader is referred to [5,6].

4. Synchronization and Sensor Calibration

For a meaningful evaluation of the recorded data, it is essential to calibrate the sensors to each other and to synchronize them in time. To avoid deviations between measurement days, the sensor system was calibrated each day before start of recording using a checkerboard and the toolboxes available in MATLAB. Both the raw data of the calibration process and the raw data of all experiments are available on request. To synchronize the radar with our reference system (lidar and cameras), a software trigger method was implemented. The reference system is synchronized internally by a HW trigger controller. The lidar sensor generates an electrical signal each time it has completed one full rotation, which is then transmitted to the camera with a delay that can be adjusted using the trigger controller. When the camera API detects the trigger event, the software trigger management thread processes the signal for each connected sensor. If a radar data acquisition thread detects the trigger event, it starts a measurement with the corresponding UDP command. The sensors of the reference system were calibrated to each other using the toolboxes in MATLAB for single as well as stereo camera and lidar-camera calibration [11,12,13]. The resulting matrices are stored in the result file, as is the translation vector between the reference system and the frontmid radar sensor.

5. Summary and Future Work

In this paper, we have presented a calibrated, rectified and synchronized dataset for radar pattern recognition and other algorithms in automotive environment with camera, lidar and radar data. This work contributes to the further development of automated and autonomous driving functions. In further work, the object recognition could be optimized and extended to other object classes. Furthermore, it would be conceivable to implement a methodology that enables object annotation with the aid of the lidar point cloud in order to create a more robust data basis for the calculation of distances and angles.

Acknowledgment

The authors gratefully acknowledge the Association of German Engineers (VDI) for financial support including funding by the German Federal Ministry for Education and Research (BMBF): KI-ASIC (Funding code 16ES0997).

References

1. Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11), 1231-1237.
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., ... & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3213-3223).
3. Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., ... & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11621-11631).
4. Déziel, J. L., Merriaux, P., Tremblay, F., Lessard, D., Plourde, D., Stanguennec, J., ... & Olivier, P. (2021, September). Pixset: An opportunity for 3d computer vision to go beyond point clouds with a full-waveform lidar dataset. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) (pp. 2987-2993). IEEE.
5. Scharf, D., Saadani, S., Schneider, M., Höß, A. (2022). A Semi-Automated Multi-Sensor Data Labeling Process for Deep Learning in Automotive Radar Environment. In 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) (pp. 1-8). IEEE.
6. D. Scharf et al. (2023). Schlussbericht zum Teilvorhaben der Ostbayerischen Technischen Hochschule Amberg-Weiden - Akquisition, Aufbereitung und Evaluation von Radar-Signalen als Datenbasis fur die Entwicklung¨ von KI-gestutzten neuromorphen Prozessorarchitekturen. Technische Informationsbibliothek Leibniz Universitat Hannover.
7. Abdulla, W. (2017). Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Github.
8. Qiao, S., Chen, L. C., & Yuille, A. (2020). DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution. arXiv 2020. arXiv preprint arXiv:2006.02334.
9. Euro, N. C. A. P. (2017). Test protocol-aeb car-to-car systems. Brussels, Belgium: Eur. New Car Assess. Programme (Euro NCAP).
10. HDF Group. (1997). Hierarchical data format version 5.
11. MATLAB, M. (2023). Camera Calibrator.
12. MATLAB, M. (2023). Stereo Camera Calibrator App.
13. MATLAB, M. (2023). Lidar-Toolbox.