The DriveU Traffic Light Dataset

A. Fregin, J. Müller, U. Kreßel and K. Dietmayer

1 Introduction

This work was developed as part of the DriveU, a research institute of Daimler and the University of Ulm. The dataset adresses to researches in the field of traffic light recognition and detection. The main contributions of this dataset are:

- Quantity: We provide the highest number of annotated traffic lights compared to existing datasets (state 2017)
- Quality: We provide accurate and consistently annotated objects due to using an appropriate labeling tool. Furthermore, all labels have novel descriptions such as viewpoint orientation, additional light information (bike, pedestrian, arrow masks) and relevancy information. Furthermore, track-ids are assigned to group unique traffic light instances.
- Additional sensor data: In addition to camera images, we provide stereo camera images, calibration data and vehicle data such as GPS and velocity and yaw rate.

More details can be found in our ICRA publication. This dataset description shall give a brief introduction into the dataset structure and the provided data. We hope to encourage other researchers in the field of traffic light recognition and are always opened for objective criticism and suggestions. If there are any uncertainties feel free to contact us.

For questions, please contact julian-2.mueller@uni-ulm.de or andreas.fregin@daimler.com

2 Dataset Overview

Table 1 compares our dataset with existing traffic light datasets. It contains four times more annotations compared with LISA [1][2], the so far largest traffic light database. Our camera sensor has a resolution of 2048x1024 pixels and the images are recorded with frame rate of 15 Hz. The images are provided with 16 or 8 bit resolution.

Table 1: Comparison of the statistics of existing traffic light datasets.

	LARA	LISA	Bosch	DriveU
Resolution [WxH]	640x480	1280x960	1280x720	2048x1024
Depth [bit]	8	8	8, 12	8, 16
Frame Rate [Hz]	25	16	16	15
Annotations	9,168	51, 826	24,242	232,039
Cities	1	1	17	11
Disparity data	✗	✗	✗	✓
Classes	4	7	15, 4	344

LARA dataset was recorded on a route in Paris, France. Also LISA dataset was recorded only recorded in one city, San Diego, USA. Bosch Small Traffic Light Dataset [3] was published in 2017 and also recorded in one region covering multiple cities. In an overall number of eleven cities different routes were recorded in order to get more variation. Recordings are divided in sequences, whereby one sequence corresponds to one intersection drive. LARA and Bosch only offer monocular camera images. LISA dataset describes their system as a stereo camera system. Unfortunately their repository does not include stereo camera images. The DriveU Traffic Light Dataset also publishes disparity images from a stereo camera. More details can be found in Section 3.

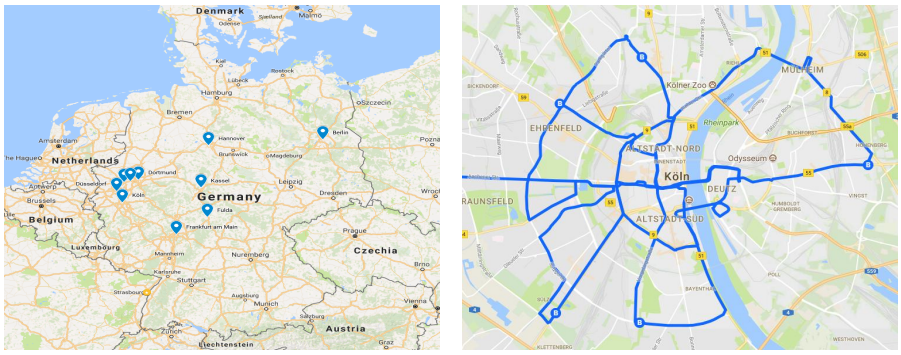


Figure 1: Recorded cities (left) and example routes in Cologne (right).

2.1 Dataset Structure

Figure 2 illustrates the database structure. The dataset is basically separated city-wise. Each city contains several routes, which are divided into sequences. The sequences contain the left unrectified camera image as well as the disparity image.

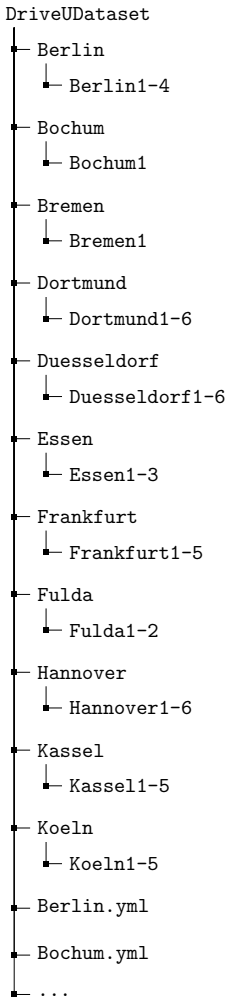


Figure 2: Structure of the DriveU Dataset. The data is separated into one folder per city. The city itself is divided into a varying number of routes. Each route is divided into sequences, whereby one sequence typically contains one intersection. The sequence folders contain all image data (left unrectified camera image and disparity image saved in a specific format). The YML-Files are in the top directory.

2.2 Detailed Overview

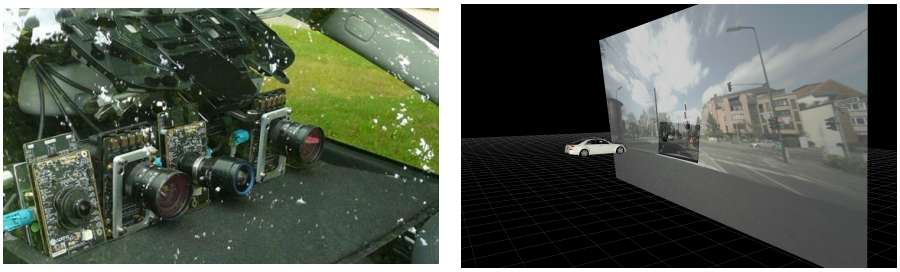
Table 2 show the recorded routes as well as the number of sequences in detail. The last column describes, whether the viewpoint orientation (front, back, left and right) is also annotated (Details see Section 5.2)

Table 2: Detailed subdivision of DriveU Dataset into recording sessions and sequences. One sequence corresponds to one intersection drive from approximately 150-200 meters up to passing the traffic light instances. Viewpoint orientation is only annotated in a subset of sequences.

City	Route	Sequences	Viewpoint orientation?
Berlin	Berlin1	43	✓
	Berlin2	68	✓
	Berlin3	56	✓
	Berlin4	34	✓
Bochum	Bochum1	41	✗
Bremen	Bremen1	34	✓
Cologne	Cologne1	86	✗
	Cologne2	72	✗
	Cologne3	57	✗
	Cologne4	51	✗
	Cologne5	68	✗
Dortmund	Dortmund1	89	✗
	Dortmund2	46	✗
	Dortmund3	63	✗
	Dortmund4	71	✗
	Dortmund5	49	✗
	Dortmund6	47	✗
Duesseldorf	Duesseldorf1	65	✗
	Duesseldorf2	50	✗
	Duesseldorf3	60	✗
	Duesseldorf4	73	✗
	Duesseldorf5	14	✗
Essen	Essen1	76	✗
	Essen2	41	✗
	Essen3	23	✗
Frankfurt	Frankfurt1	55	✗
	Frankfurt2	54	✗
	Frankfurt3	51	✓
	Frankfurt4	73	✗
	Frankfurt5	35	✓
Fulda	Fulda1	12	✓
	Fulda2	23	✓
Hannover	Hannover1	79	✗
	Hannover2	69	✗
	Hannover3	56	✗
	Hannover4	47	✗
	Hannover5	37	✗
	Hannover6	64	✗
Kassel	Kassel1	15	✓
	Kassel2	38	✓
	Kassel3	31	✗
	Kassel4	47	✓
	Kassel5	20	✗

3 Sensor Data

The chosen sensor setup for traffic light recognition is purely camera-based. Figure 5 illustrates the used cameras (left) and their viewing-angles (b). The camera setup consists of a stereo camera pair and two wide-angle monocular cameras. The wide-angle cameras are not explained in detail as they are not published in the DriveU Traffic Light Dataset.



(a) Stereo and wide-angle cameras

(b) Viewing-angle comparison

Figure 3: Camera setup and viewing-angle comparison between both cameras. . The highlighted cutout shows the left stereo camera image overlapped with the wide-angle frame.

3.1 Stereo Camera Images

The stereo cameras have a CMOS sensor with a resolution of 2048 pixels in width and 1024 pixels in height. The raw images are saved with a bayer-pattern (GB) in 12 bit depth, whereas 16 bit depth is obtained by correcting the 12 bit raw image according to the sensor characteristic curve. Figure 4 illustrates one example scene recorded by the stereo camera as left and right image. The published data only contains the unrectified left camera image.

(a) Left unrectified camera image

(b) right unrectified camera image

Figure 4: Left and right stereo camera images. Both images are not rectified.

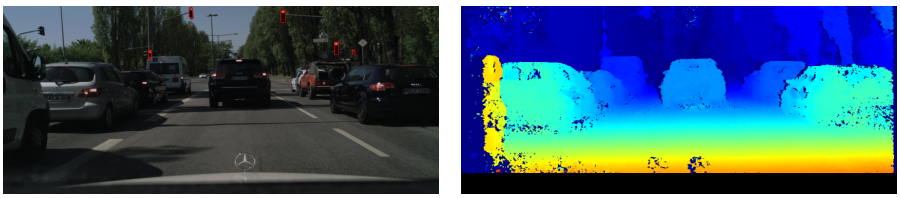
3.2 Disparity Images

As already mentioned, the Daimler Traffic Light Dataset also provides disparity images. Therefore, the SGM [4] algorithm with modifications of Gehrig et al [5] is used. This implementation is adapted to typical vehicle environment on (urban) roads. When working with these disparity images, there a several points that have to be noted:

Resolution The color images have a resolution of 2048x1024 pixels. The disparity image has a resolution of 1024x440 pixels. This is caused by time constraints. The disparity image is calculated in the image cycle (15 Hz). In order to guarantee real-time capability a disparity calculation on the high image resolution is not possible.

Mapping Calculating a 3D position of an object in the color image, the disparity value has to be known. Therefore, a first rectification step is necessary (rectify the bounding box coordinates). In a second step, the different resolution factor has to be considered (e.g. pixel (100,100) in the rectified color image corresponds to pixel (50,50) in the disparity image). To avoid this mapping step, the disparity image can be upsampled. However, this method does not lead to information acquisition but way higher computational effort.

Absolute Disparity Values As already mentioned, the disparity calculation is done on a smaller resolution. Therefore, the original images are also resized to 1024x440 pixels and rectified before. This means, the absolute disparity values are related to this image resolution. When working with the larger left camera image together with the small disparity image, the absolute disparity value has to be doubled. Another possibility would be to change the calibration matrices by the binning factor, which is way more complex.



(a) left camera imge (b) corresponding disparity image

Figure 5: Left camera image (a) and corresponding disparity image (b) in color visualization. Red and yellow colors are close (high disparity), whereby blue shows far objects (small disparity).

Parameters like focal length and baseline, which are necessary for disparity to depth conversion can be found in the calibration data.

3.3 Vehicle Data

In addition to sensor and label data a small subset of vehicle data is published. The following vehicle data are published

- **Velocity** v of the vehicle measured in meters/second $[\frac{m}{s}]$
- **Yaw-Rate** $\dot{\psi}$ of the vehicle measured in radian/second $[\frac{rad}{s}]$
- **Longitude** λ (GPS) in degree [°]
- **Latitude** ϕ (GPS) in degree [°]

The yaw-rate is given counterclockwise (see Figure 6).

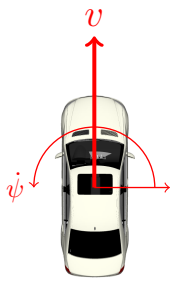


Figure 6: Velocity and yaw-rate. Yaw-rate is given counterclockwise.

Velocity and Yaw-Rate are updated each image cycle, whereby the GPS position is only updated with 1 Hz.

4 Calibration

All cameras we used for recording are intrinsicly and extrinsicly calibrated.

4.1 Coordinate Systems

horizontal-traffic-lights-595x287 (1) Two main coordinate systems are defined:

- Vehicle Coordinate System: X front, Y left and Z up
- Camera Coordinate System: X right, Y down, Z front

The vehicle coordinate system is located at the rear axis. The camera coordinate system is located behind the windshield (see Figure).

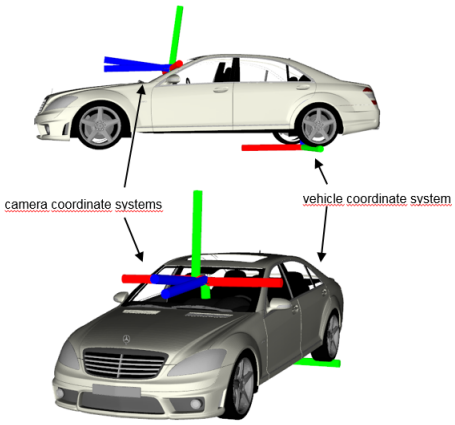


Figure 7: Coordinate system definitions: The vehicle coordinate system is located at the rear axis. The camera coordinate systems are behind the windshield.

4.2 Intrinsic Camera Matrix

The intrinsic camera matrix is provided as

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

with the focal length f and principal point c in x- and y-direction, respectively.

4.3 Projection Matrix

The projection matrix is provided as

$$P = \begin{bmatrix} f'_x & 0 & c'_x & T_x \\ 0 & f'_y & c'_y & T_y \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2}$$

with the focal length f and principal point c in x- and y-direction, respectively. T_x is is given as $T_x = -f'_x \cdot B$, where B is the baseline of the stereo camera.

4.4 Rectification Matrix

The rectification matrix is provided as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3}$$

4.5 Distortion Matrix

The distortion matrix is provided as

$$D = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & k_3 \end{bmatrix}, \tag{4}$$

where k_1, k_2 and k_3 are radial distortion and p_1 and p_2 are tangential coefficients.

4.6 Extrinsic Matrix

The extrinsic matrix is provided as

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}, \tag{5}$$

composed of rotation and translation part. Please note, that this matrix gives the transformation from the vehicle rear axis to the left stereo camera frame in ZYX rotation order.

5 Label Rules and Class Identity

Two requirements concerning the annotation of our dataset

5.1 Label Tool

All annotation of the DriveU dataset are annotated in our labeling tool. Actually, traffic lights are annotated with rectangular bounding boxes. After creating new traffic light label, it can either be assigned to an existing traffic light instance or a new instance can be created. In an additional window, the created bounding box is shown in a zoom visualization. Thereby, very accurate boxes can be created. This is especially important for distant objects with a small resolution. Traffic lights are labeled up to the resolution limit.
Our recorded data is divided into cities and sequences. One sequence is one drive at an intersection up to passing the traffic lights. Sequences are labeled from their end to the start. This way objects are big in the beginning and small in the end, which helps to ensure no object is overseen because of a tiny size and generally results in a better degree of completeness. Furthermore, state estimation is better even for small objects.
In order to describe the properties of a traffic light label tags are used. The combination of tags leads to a specific class identity explained in the next subsection. To avoid labeling errors, the tags are monitored and invalid combinations of properties (e.g. red and green traffic light states simultaneously) are

directly displayed to the labeler. Missing tag properties are also displayed.

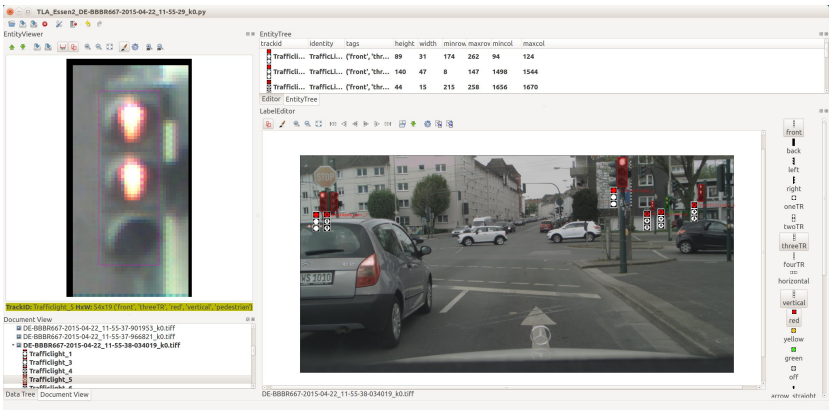


Figure 8: Graphical User Interface of the traffic light labeling tool. Zooming option as well as several property tags are available.

5.2 Tags and Class Identity

As already mentioned the class identity describes different properties of a traffic light label. Following properties (tags) are specified:

- **Viewpoint orientation:** front, back, left, right
- **Number of lights:** one, two, three, four
- **Installation orientation:** horizontal, vertical
- **State:** red, yellow, green, off
- **Light mask:** arrow straight, arrow left, arrow left 45, arrow right, arrow right 45, bicycle, pedestrian, tram
- **Occlusion**
- **Relevancy**

Viewpoint orientation is a property, which we defined only recently during labeling process. Thus, only a subset of the dataset contains viewpoint orientation information (see Table 2). It is especially interesting for training convolutional neural networks in order to separately train front traffic lights (relevant for the vehicle, state determination possible) from back traffic lights, which are not important for the vehicle. Furthermore, negative sample creation from back traffic lights is avoided (which inevitable happens for random negative sampling with maximum IoU overlap).
Installation orientation mainly depends on the country-specific laws. As DriveU

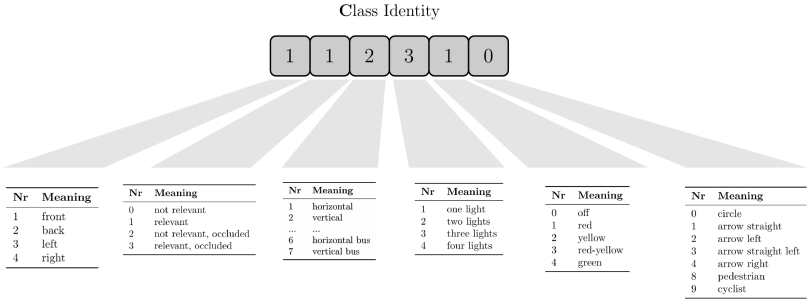


Figure 9: Class identity of a DriveU dataset object. It consists of 6 digits expressing properties.

dataset only consists of recordings in Germany, only vertical traffic lights exist. The state tag represents the lamp of the traffic lights. Only one state can be tagged, with the exception of red and yellow together, which expresses a toggle process from red to green. In contrast, a toggle process from green to red is indicated by yellow only. Red-yellow traffic lights to not exist in each country. Traffic light lamps exists in a various number of shapes. The most common shape is circular. Turning lanes are typically equipped with arrow-shaped traffic lights. Pedestrian, bicycle and bus/tram traffic lights are also annotated.
Preceding vehicles often occlude the view to the traffic lights. Occluded objects are tagged.
A particular labeled property is the relevancy-tag. Intersections contain traffic lights, which are relevant for the current route of the vehicle. Other traffic lights are not relevant for the actual route. We tag relevant traffic lights in our dataset to allow evaluation on relevant objects only.

Finally, all tags are combined to a class identity, which consists of six digits. Figure 10 illustrates the meaning of each digit. The first digit summarizes the viewpoint orientation. Relevancy and occlusion are represented by the second digit. The third digit expresses installation orientation. The fourth digit gives the number of lights, the fifth the color or state. The last digit expresses the light mask shape.

Figure 10: Relevant traffic lights (red) and not relevant traffic light (white). Relevant relates to the planned route and the actual lane. More details and discussion see [6].

6 Tooling

We provide classes and methods in C++, Python and MATLAB to load the images, annotation files and calibration data. The following figure illustrates the structure of the tooling repository. The C++ repository defines relevant classes for the overall database, an image class, an object (label) class, a vehicle data class as well as a calibration data class. The database contains a list of images, whereby each image contains a list of labeled objects and the vehicle data. The C++ code provides many functionalities in to illustrate the color image, the labeled image and the disparity image as a cv::Mat. If OpenCv is not installed, they can not be used. Calibration matrices can be returned either as cv::Mat, or OpenCv independent. How to use the classes and methods can

be seen by using the test or main functions, respectively.

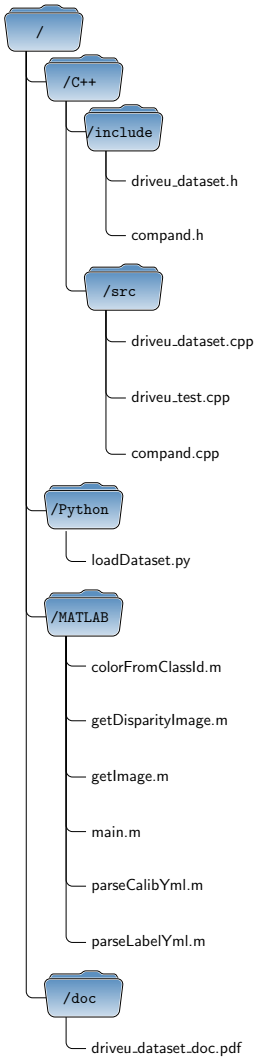


Figure 11: Structure of the DriveU Dataset Tooling repository. We provide dataset parsing in C++, Python and MATLAB.

7 Citation

When using the DriveU Dataset for your scientific work we would appreciate to be cited by you.

DOI: `https://doi.org/10.1109/ICRA.2018.8460737`

References

[1] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, 2016.

[2] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," in *IEEE 18th International Conference on Intelligent Transportation Systems, ITSC 2015, Gran Canaria, Spain, September 15-18, 2015*, pp. 2341–2345, 2015.

[3] K. Behrendt and L. Novak, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, IEEE.

[4] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.

[5] S. K. Gehrig, F. Eberli, and T. Meyer, "A real-time low-power stereo vision engine using semi-global matching," in *Computer Vision Systems, 7th International Conference on Computer Vision Systems, ICVS 2009, Liège, Belgium, October 13-15, 2009, Proceedings*, pp. 134–143, 2009.

[6] A. Fregin and K. Dietmayer, "A closer look on traffic light detection evaluation metrics," in *19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016, Rio de Janeiro, Brazil, November 1-4, 2016*, pp. 971–975, 2016.