Developement of an autonomous driving environment model visualization based on object list level

Tobias Wagner Christoph Zach Max Haindl Philipp Korn Stehpan ...? Dennis Roessler Max Pfaller Domi ...?

***Abstract*— This electronic document is a live template. The various components of your paper [title, text, heads, etc.] are already defined on the style sheet, as illustrated by the portions given in this document.**

I. INTRODUCTION

II. RELATED WORKS

IEEE Fabio Reway Test Method for Measuring the Simulation-to-Reality Gap of Camera-based Object Detection Algorithms for Autonomous Driving

III. MATERIALS AND METHODS

Kleines Intro was jetzt kommt

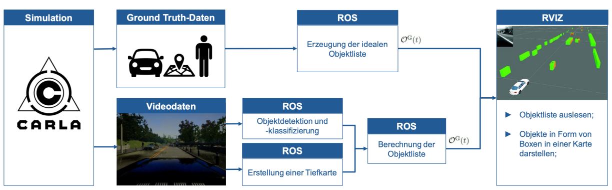

Fig. 1. Ueberblick

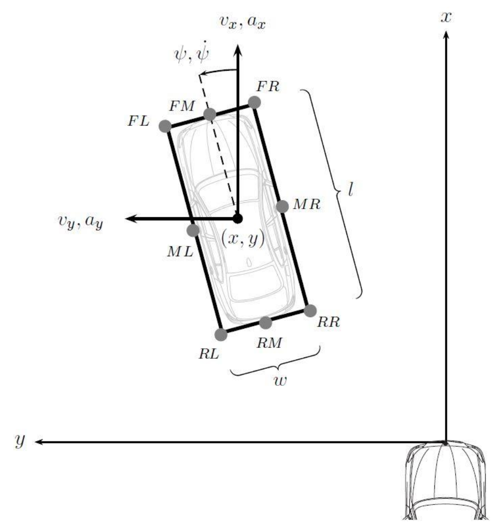

Fig. 2. Fahrzeugkoordinatensystem

A. Creating simulation scenario

- Welcher Simulator wurde verwendet
- Welches Szenario (NCAP)
- Szenario beschreiben

B. Creating objects list of ground-truth data (TP1

- Erstellung Objektliste
- Objektliste anhand Attribut-Vektor beschreiben
- Ros-System beschreiben
- Feature Vektor Ermittlung

C. Evaluation of video data (TP2)

- Detektion Objekte (Yolo)
- Tracking Objekte (Tracker)
- Gleichung zur Berechnung von zB Geschwindigkeit, Beschleunigung
- Ermittlung Classification / prop mov / prop exis

D. Visualization of object lists(TP3)

- subscription der Objektlisten
- Auswertung der Objektlisten,Marker Array, Tf Transform,

E. Evaluation of object lists(TP3)

- Objektlisten in BagFiles aufnehmen
- Geo und Time mapping
- Berechnung von iOu etc. verweis auf Veröffentlichung von Fabio

*This work was not supported by any organization

F. *Evaluation of object lists - Post processing*

After recording object list data streams in Rosbag files there is the possibility to analyse data in different ways by the post processing application offline. The user can either analyse single data streams or compare two different ones. An essential usage would be the comparison of a simulation data stream (Ground-Truth data) to a sensor data stream (camera data). The post processing application provides the opportunity to display several data values within the recorded Rosbag file relating to contained message frames.

1) Basic analysis: Considered to one single Rosbag file specific attribute values of single objects which are selected by their object ID can be displayed. The variety of available attribute types is shown in table . In addition, the number of detected objects can be visualized.

2) Advanced analysis: Regarding two Rosbag recordings further analysis methods for comparing the streams are provided. In this case a common time base needs to be generated. To avoid errors because of time variation of both recordings following mapping algorithm is executed. Each frame time stamp is handled as time relative to its Rosbag start time in milliseconds. To every frame in the Rosbag file which is provided by sensor data a frame of simulation data is dedicated. The simulation frame to choose is the latest past frame in relative stream time. The principle of frame mapping is shown in figure 3. With this mechanism pairs of frames are generated (sensor frame with corresponding simulation frame).

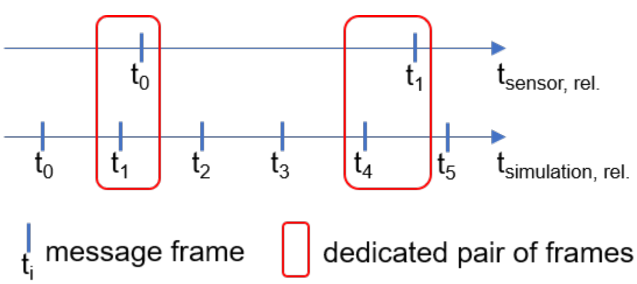

Fig. 3. Principle of frame mapping algorithm

A fundamental use case in analysing the recorded Rosbag files is to evaluate the quality of sensor data. Therefore, it is necessary to operate the mapping of object detection by the sensor in comparison to the simulation data. To determine whether a given camera object is evaluated as True Positive (TP), False Positive (FP), False Negative (FN) or a mismatch (mm), the **Intersection over Union** (IoU) value is used, like shown in [**?**]. Like shown before, in general there is a list of m camera objects (B_{pr}) and a list of n ground truth objects (B_{gt}) for each frame. To evaluate a frame, for each combination of GT object and camera object the IoU value is calculated. All those values build a matrix like shown in Table I.

TABLE I

IoU-Matrix for a single frame

$IoU(B_{gt,1},B_{pr,1})$	$IoU(B_{gt,2},B_{pr,1})$	...	$IoU(B_{gt,n},B_{pr,1})$
$IoU(B_{gt,1},B_{pr,2})$	$IoU(B_{gt,2},B_{pr,2})$	...	$IoU(B_{gt,n},B_{pr,2})$
...	...	...	...
$IoU(B_{gt,1},B_{pr,m})$	$IoU(B_{gt,2},B_{pr,m})$	...	$IoU(B_{gt,n},B_{pr,m})$

A given camera object $B_{pr,i}$ is ...

... **FP**, if there is no value

$$IoU(B_{gt,k},B_{pr,i}) > t \quad \text{with} \quad k \in \{1,...,n\} \tag{1}$$

in the according row of the matrix which is greater than the given threshold.

... **FP**, if there is one or more IoU values in the according row greater than the threshold, but for every $B_{gt,k}$, for which equation 1 is true, there is another $B_{pr,j}$ $(j \neq i)$ which matches with $B_{gt,k}$ and $IoU(B_{gt,k},B_{pr,j}) > IoU(B_{gt,k},B_{pr,i})$

... a mismatch (**mm**) if it is no FP case, but none of the found possible matching $B_{gt,k}$ has the same class as $B_{pr,i}$.

... **TP**, also called a match, if none of the other mentioned cases are detected. That means, that there is at least one $B_{gt,k}$ which fulfills equation 1 and has the same classification as $B_{pr,i}$ and there is no other $B_{pr,j}$ which matches better with the found $B_{gt,k}$

Going through the rows of the matrix, for each $B_{pr,i}$ in the given frame it can be decided, whether the case is TP, FP or mm.

On the other way round, examining the Ground Truth objects $B_{gt,k}$, that means the columns of the calculated matrix, all FN cases can be detected. It is an **FN**, if there is no $B_{pr,i}$ for which

$$IoU(B_{gt,k},B_{pr,i}) > t \quad \text{with} \quad i \in \{1,...,m\} \tag{2}$$

Going through the columns of the matrix, this decision can be made for every $B_{gt,k}$.
With this steps, a given frame with m camera objects and n ground truth objects can be investigated.

In this project these functions, one for investigating the camera objects and one for detecting all FN cases, were realized in Python. The calculation of an IoU value is processed with functions of the package *shapely* [**?**]. First, the given objects, which are defined through their properties x, y, *length*, *height*, *yaw* and *classification* like presented in [**?**] are transformed into bounding boxes with a shapely function. With two of these bounding boxes shapely can calculate the intersection area and the union area, and so the IoU value can be processed.

As a result of object mapping it is possible to visualize the number of True Positive (TP), False Positive (FP), False Negative (FN) and mismatch (mm) cases per frame. Further, following quality of service parameter according to **??** can be displayed:

- recall per frame
- precision per frame
- FPPI per data stream (sensor)
- MOTA per data stream (sensor)
- MOTP per data stream (sensor)

Another feature of the post processing application is the analysis of deviations by calculating differences of specific attribute values between two recorded data streams. For that matter only True Positive cases in object mapping are regarded and the concerning object is selected by its object ID in the simulation data record. The difference value results from

$$difference = value_{simulation} - value_{sensor} \tag{3}$$

On top of every post processing analysis - except from FPPI, MOTA and MOTP - the mean value and standard deviation of the corresponding value series are calculated.

3) Graphical User Interface: To investigate the quality of the processed camera object data, a graphical user interface (GUI) was created. It was designed with Pythons binding package for Qt (PyQt5) [**?**] and defined as a plugin for *rqt*, a *ROS* framework for GUI development [**?**]. With this plugin, the user can import two bag files, one ground truth data bag file and one camera data bag file. By using the functions mentioned before, the GUI can show several data graphs to the user, like raw data plots, comparing plots with object data of both files or evaluation data. Along with the data the mean value and the standard deviation for each data set is portrayed in the GUI.
Apart from data plots the interface can also show quality parameters for the whole camera data bag file in an extra widget. For each operation, where IoU calculation is needed, the user can set the threshold value for the evaluation.

4) Results: The performance of processed camera data can be evaluated with metrics presented in [**?**], which are realized like introduced in part III-F.2. For the given scenario the reached performance is shown in Table II.

TABLE II
PERFORMANCE RESULTS

threshold	$t = 0.5$	$t = 0.6$	$t = 0.7$
Precision	...	...	...
Recall	...	...	...
FPPI	...	...	...
MOTA	...	...	...
MOTP	...	...	...

5) Conclusions:

IV. RESULTS

Ergebnisse des Projekts:

- Funktioniert die Auswertung
- Wie gut sind die Kamerawerte im Vergleich zu Groundtruth Werte
- sind die Werte represenativ

V. CONCLUSIONS

APPENDIX

ACKNOWLEDGMENT