Abstract—In this paper, it is shown that synthetic images can be used to test specific use cases of a lane tracking algorithm which has been developed by Audi AG. This was achieved by setting up a highly configurable and extendable simulation framework “Virtual Test Drive”. The main components are a traffic simulation, visualization and a sensor model which supplies ground truth data about the street lanes. Additionally, the visualization is used to generate synthetic camera sensor data. The testbed also contains a realistic driving dynamics simulation and a real image processing soft ECU (which is represented as a standard PC in the early development stages). One of the modules on the image processing ECU is a lane tracking algorithm. The algorithm is designed to calculate the transition curves while driving. This information can be used as input for driving assistance functions, e.g. lane departure warning. By running the lane tracker on a synthetic image it is possible to compare the results of the lane tracker with the ground truth data provided by the simulation. In this particular case, the information has been used to test and optimize parts of the systems by using specific and determined scenarios in the simulation.

Index Terms—image processing, testing, synthetic data, simulation

I. INTRODUCTION

As cameras and other sensors become increasingly common in modern production cars, image processing allows new functions to improve comfort and safety. Popular examples are pedestrian or vehicle detection systems to avoid collisions but also lane detection to help the driver stay on track.

Those algorithms are conventionally tested with real sensor data and their results are validated with ground truth. However, ground truth is difficult to measure. For example video data for a road tracker would need to be labeled with the exact clothoid parameters of roads and road marks, respectively. Creating those labels is very time-consuming and expensive. For these reasons generating synthetic images to simulate vision-based algorithms is of raising importance [5]. The paper is organized as follows: Sec. 2 gives an overview of related work; Sec. 3 introduces the architecture and components of the used testbed; Sec. 5 and 6 present the used lane tracking algorithm and the methodology behind; and in Sec. 7 the results of the presented approach are summarized.

II. RELATED WORK

The complete simulation framework used in this paper was introduced in [1]. In [6] this simulation framework has first been used to optimize specific parameters of a lane tracker algorithm with synthetic images as input generated by the simulation toolchain. The lane tracker is based on algorithms described in [4]. A similar test setup was established in [5], however, the focus of the work was testing a lane tracking algorithm in combination with different levels of artificially created image noise. This paper focuses on the estimation quality of four specific output parameters of a lane tracking algorithm on seven distinct road designs.

III. TESTBED ARCHITECTURE

The Virtual Test Drive (VTD) simulation framework allows users to fulfill a wide range of different simulation tasks throughout the entire development process of driver assistance and active safety systems. Depending on the exact use case of the tasks, different types of simulators can be used [1].

In this case, a PC-based VTD-simulator is connected to a hardware-in-the-loop (HIL) simulator and an image processing electronic control unit (ECU). Additional electronic automotive systems, such as ECUs, sensors, and actuators, can be connected to the HIL-simulator. With this extended system it is possible to test complete functions from top to bottom, e.g. a lane departure warning lamp can be activated as soon as the ego car is crossing a lane inside the virtual environment.

![Figure 1. Testbed Architecture](image)
The implementation of the lane tracker is based on the “4D-approach” described by [4] and has been developed by the Audi department for Advanced Driver Assistance Systems. The steps of the general tracking procedure are only described briefly here; for more detailed information on the underlying models the reader is deferred to [3].

The center of a lane is modeled as a transition curve whose curvature at distance $l$ along its pathway is described by the equation [6]

$$c(l) = c_0 + c_1 \cdot l$$

where $c_0 = \frac{1}{r_0}$ is the curvature of the circle with radius $r_0$ in the starting point of the transition curve. Parameter $c_1$ describes the change in curvature along the pathway. To calculate the expected positions of lane markers, bordering the own lane, the width $B$ of the lane must be taken into account. In order to compensate for the relative position of the ego vehicle towards the position of the transition curve’s origin some other parameters are estimated:

- pitch angle $\theta$ of the vehicle’s x-axis towards the ground plane,
- yaw angle $\omega$ of the vehicle’s x-axis towards the tangent to the transition curve at its origin,
- lateral offset $Y_{off}$ of the vehicle towards the origin of the transition curve.

In each iteration of the tracking loop the estimated state vector $\hat{x}_{k-1}$ is predicted to the current time step $k$ using a transition matrix. Using the predicted state vector $\hat{x}_{k}$ the lane center transition curve model is calculated and support points along the pathway are created. For each support point the expected points on the left and the right lane border are produced by moving a distance of $\pm \frac{B}{2}$ along the perpendicular to the tangent of the transition curve model at the particular support point. In a next step these points are projected to the coordinate frame of the imaging chip using the known camera pose within the vehicle and a pinhole camera model. The coordinate transformation results in pixel positions where edge points of lane markers are expected and can be looked for with an adaptive convolution mask.

In the update step of the Extended Kalman Filter [7] the measurement residuals $(z_k - H_k \hat{x}_{k})$ are determined, weighted by the Kalman Gain $K$ and added to the last predicted state vector [7]:

$$\hat{x}_k = \hat{x}_{k} + K (z_k - H_k \hat{x}_{k})$$

Thus, the prediction for the next time step can be performed. For simplicity reasons the equations for the computation of the error covariance matrix are omitted here, these can be found in [7]. An example of the lane tracker working on real and rendered synthetic images can be seen in Figure 2 and Figure 3.

**V. Methodology**

The following paragraph describes a testing methodology which was used to evaluate the lane tracker’s estimation of
the lane width $B$, lateral offset $Y_{off}$, the curvature $c_0$ and the first curvature derivative $c_1$ using the simulation environment.

Therefore seven simple test scenarios which consist of a road with road marks only were created. In every scenario one or maximum two parameters have been changed to be able to compare the road tracker result directly with scenario’s ground truth data.

The 7 different test scenarios (examples in Figure 4)

1) basic scenario: $B = 3m$, $Y_{off} = 0m$, $c_0 = 0$, $c_1 = 0$ and the cars velocity $v = 50km/h$ (all parameters were used as default for the other scenarios)

2) wide lane: $B = 6m$

3) narrow lane: $B = 2m$

4) lateral offset left: $Y_{off} = 1m$

5) lateral offset right: $Y_{off} = -1m$

6) little curvature: $c_0 = \frac{1}{5000m}$, $v = 20km/h$

7) strong curvature: $c_0 = \frac{1}{500m}$, $v = 20km/h$

The road tracking algorithm was used on all scenes under the assumption of a constant vehicle velocity. This implementation of the algorithm did not use any initialization routine, so it takes a few seconds to engage to the correct approximation of the parameters. Thus, preliminary data until complete transient oscillation were not used for the evaluation.

Figure 5 shows the estimated width of scene 3. It shows that it takes approximately 120 frames for completing transient oscillation.

VI. RESULTS

The values of the estimation parameters have been aggregated over all scenario frames. Table 1 shows the results using following legend:

- **Breite** is width in meters
- **Ablage** is lateral offset
- **Krummung** is curvature
- **Krümmungsänderung** is curvature derivative
- **Gr.-Truth** is the ground truth data that was used in the simulation
- **Mittel** is the average value over all frames
- **Fehler** is the difference between Gr.-Truth and Mittel
- **Std.-Abw.** is the standard deviation

It can be seen that the estimated parameters were very accurate. The error in lane width is almost always less than 1 cm, only in scene 7 it is 1.3 cm. The error in lateral offset is also less than 1 cm, except in scene 2 where it is 1.2 cm. The curvature was estimated absolutely correct in all scenes but scene 7 which was using a relatively high curvature. The curvature derivative was set to zero in all scenes and the estimated parameters were never above $10^{-4}$ which is a tolerable error.

VII. CONCLUSION

The Virtual Test Drive simulation environment combined with an image processing algorithm simulator offers the application engineer a unique chance to test and analyze individual parameters on determined test scenarios. It is now possible to
do functional trials in a closed loop set-up. This approach can help to decrease the amount of real road trials and thereby to save costs and to reduce risks for man and machine, respectively.

The use case described in this paper demonstrates the potential of this simulation testbed. The results proved that it is possible to test specific output parameters of image processing algorithms using synthetic image data. The major benefit of this testbed is the opportunity to automate parameter studies using ground truth data and a weighting function to measure the quality of the results. Nonetheless, it is always necessary to validate the adaptations in real road trials.

REFERENCES


<table>
<thead>
<tr>
<th>Sensor</th>
<th>Berechnung [m]</th>
<th>Ablage [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>φ</td>
<td>Gr.- Anl.</td>
<td>Mittel</td>
</tr>
<tr>
<td>1</td>
<td>3,000</td>
<td>2,950</td>
</tr>
<tr>
<td>2</td>
<td>3,000</td>
<td>2,950</td>
</tr>
<tr>
<td>3</td>
<td>3,000</td>
<td>3,000</td>
</tr>
<tr>
<td>4</td>
<td>3,000</td>
<td>3,000</td>
</tr>
<tr>
<td>5</td>
<td>3,000</td>
<td>3,000</td>
</tr>
<tr>
<td>6</td>
<td>3,000</td>
<td>3,000</td>
</tr>
<tr>
<td>7</td>
<td>3,000</td>
<td>3,000</td>
</tr>
</tbody>
</table>

Table I

RESULTS OF THE PARAMETER TESTS