CASE STUDY

Technische Hochschule Deggendorf

Faculty of Applied Computer Science

Master of Automotive Software Engineering

Lecture: Advanced Driver Assistance Systems (ADAS)

Topic: Camera and CAN

Name:
Deep Bharatbhai Savaliya

Matriculation number:
12501180

Supervisor:
Prof. Thomas Limbrunner

Submission date: Deggendorf, 01.07.2025

I declare that I have written this thesis by myself and have only used the sources and aids mentioned, and that I have marked direct and indirect citations as such. This thesis has not been submitted prior for any other examination.

I agree that the results of this study work / thesis may be used free of charge for research and lecturing purposes.

Deggendorf, 01.07.2025 Deep Bharatbhai Savaliya

Camera and CAN Integration in Advanced Driver Assistance Systems in Advanced Driver Assistance Systems (ADAS)

Table of Contents

1. Introduction

Advanced Driver Assistance Systems (ADAS) play a crucial role in enhancing road safety, comfort, and automation capabilities in modern vehicles. This case study addresses two fundamental components of ADAS: automotive camera systems and the Controller Area Network (CAN) communication protocol. Cameras are increasingly deployed for object detection, lane tracking, and driver monitoring, while CAN serves as a real-time backbone for communication among Electronic Control Units (ECUs). The aim of this work is to explore the technical foundations of automotive cameras and CAN systems, including functional principles, signal processing, and feasibility of integration. This includes a deeper look into stereo vision mathematics, camera sensor specifications, signal scaling from PDU frames, convolution filtering of images, and the evaluation of data transmission via CAN at Layer 2.

Furthermore, the IEEE paper titled "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey and Taxonomy" is summarized to align this work with current research directions.

The structure of this document is as follows:

- Section 2 provides detailed material and methodology, including theoretical foundations and practical implementations.
- Section 3 presents calculated or derived results.
- Segment 4 talks about the suggestions and impediments of the findings.
- Section 5 concludes with an outline and viewpoint on future headings.

1.1. Introduction of Camera Sensor Technology in Automotive Applications

Automotive cameras operate on the principle of capturing light through a lens and projecting it onto an image sensor (typically CMOS), which converts the light into electrical signals. These signals are digitized and processed to extract visual information relevant to driving assistance, such as lane markings, obstacles, traffic signs, or pedestrians.[1]

1.1.1 Core Properties

- Field of View (FOV): Varies depending on application—wide for surround-view, narrow for long-range detection.
- Resolution: Determines detection clarity and classification precision (e.g., 1.2MP to 8MP).
- Dynamic Range: High Dynamic Range (HDR) sensors capture scenes with large brightness differences (e.g., sunlight and shadows).
- Frame Rate: Typically, 25–60 fps, affecting real-time responsiveness.
- Interfaces: CSI-2 and LVDS for data transfer; SPI/I^{2}C for configuration.

1.1.2 Environmental Influences

- Rain, fog, snow can obscure lens or reduce image quality
- Lens glare and exposure challenges in high-contrast scenes
- Temperature variations affect sensor performance and optics alignment

1.2. Introduction of CAN Communication in Automotive Applications

1.2.1 Core Properties

- Bus Type: Linear bus topology with termination at both ends
- Bit Rates: Up to 1 Mbit/s (CAN 2.0B); newer versions like CAN FD support higher payloads
- Message Format: ID-based, with 8-byte payloads (CAN 2.0) or more (CAN FD)
- Arbitration: Priority-based using CSMA/CA (collision-free)

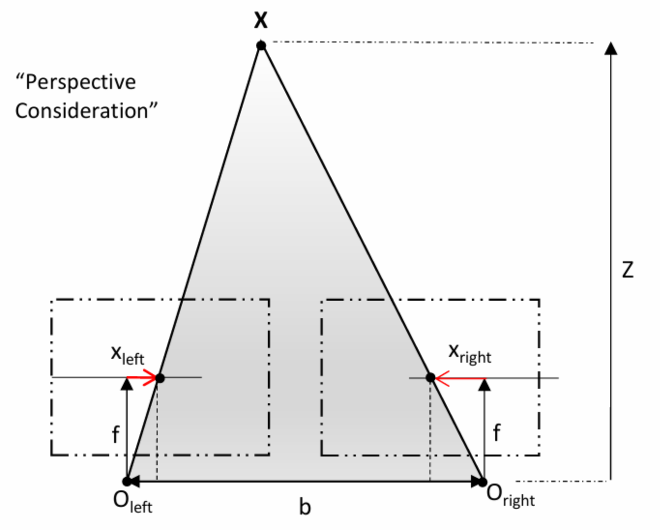
1.2.2 Environmental Influences

- Signal integrity may degrade due to reflections or poor termination
- EMC (electromagnetic compatibility) requirements must be met for stable operation
- Wire harness damage or connector corrosion can cause bus errors

2 Material and Methodology

2.1 Distance Measurement Using Stereo Camera – Disparity

A stereo camera system uses two horizontally aligned cameras (left and right) with a known baseline distance (b) between them. Both cameras observe the same 3D scene, but the same object appears at different horizontal positions (x_left, x_right) in each image due to their different viewpoints. This horizontal shift is called disparity (d). [1]



d = x_{left} - x_{right} = $\left(\frac{f\,.\,X_r}{Z_r} + \frac{f\,.\,b}{Z_r}\right) - \frac{f\,.\,X_r}{Z_r}$

d = $\frac{b\,.\,f}{Z}$

With **d** as the metric disparity of rectified stereo images, it follows for the **spatial coordinates** (X, Y, Z): [5]

X = $\frac{X_{left} \cdot\, b}{d}$

Y = $\frac{Y_{left} \cdot\, b}{d}$

$$\mathbf{Z} = \frac{b\,.\,f}{d}$$

Z = Distance to the object (depth)

f = Focal length of the lens

b = Baseline (distance between left and right camera centers)

d = Disparity, i.e., pixel shift between corresponding points in the two images [2]

EX.01 - Let's assume a stereo camera system is mounted on the front of an autonomous vehicle with the following known parameters:

Baseline b = 12 cm = 0.12 meters

Focal length f = 1000 pixels (obtained from camera calibration)

The stereo image pair shows a known object (e.g., a stop sign), and its horizontal position in the left and right images is:

- x_{left} = 560 pixels
- x_{right} = 530 pixels

So, the disparity d is:

$$d = x_{\text{left}} - x_{\text{right}} = 560 - 530 = 30 \text{ pixels}$$

Now calculate the depth (Z) of the object:

$$Z = \frac{\boldsymbol{b \,.\, f}}{\boldsymbol{d}}$$

$$Z = \frac{\mathbf{0.12\,.\;1000}}{\mathbf{30}} = \frac{120}{30} = \mathbf{4.0}\ \boldsymbol{Meters}$$

2.2 CAN Layer 2: Net Data Rate and Propagation Time

2.2.1 Maximum Net Data Rate (Without Higher Protocol Layers)

The net data rate at Layer 2 refers to the proportion of transmitted bits that represent actual payload data, excluding protocol overhead such as control, identifier, CRC, and ACK fields. This is critical in embedded automotive systems where bandwidth is constrained. [3]

R_{gross} : Gross bitrate – the raw transmission rate in bits per second (bps), (e.g., 500 kbit/s or 1 Mbit/s.)

R_{net}: Net data rate – the effective rate of payload data transmission.

$R_{payload}$: Number of payload bits (actual user data).

D_{total}: Number of total transmitted bits in one CAN frame (including all overhead).

A typical full CAN frame (with no stuffing or errors) includes:

Field	**Bits**
SOF (Start of Frame)	1
Arbitration Field (ID + RTR)	12
Control Field (IDE + DLC etc.)	6
Data Field (max)	0–64 bits (8 bytes × 8)
CRC (15 + delimiter)	16
ACK (2 bits)	2
EOF (End of Frame)	7
Total (Max payload)	**~130 bits**

The efficiency η of the protocol is the ratio of data bits to total bits transmitted:

$$\eta = \frac{D_{payload}}{D_{total}}$$

Now, to find the net data rate, we multiply the gross rate by the efficiency:

$$R_{net} = \eta \,.\, R_{gross}$$

$$R_{net} = \frac{D_{payload}}{D_{total}} \,.\, R_{gross}$$

EX.02 - Example Using CAN2.0

Max payload: $D_{payload}$ = 64bits (8 data bytes)

Typical full frame: D_{total} ≈ 130bits

So: $\eta = \frac{64}{130} \approx 0.492$

Then:

For $R_{gross} = 500kbit/s$:

$$R_{net} = 0.492\,.500 = 246kbit/s$$

For $R_{gross} = 1Mbit/s$:

$$R_{net} = 0.492\,.1000 = 492kbit/s$$

2.2.2 Propagation Time per Bit

Requirement: Signal propagation time T_d negligibly small compared to step size (bit time) T_{Bit}.

The bitrate (or baud rate) tells us how many bits per second are being sent. So, the time to send 1 bit is:

If 1,000,000 bits are sent in 1 second, then 1 bit takes $\frac{1}{1000000} = 1\mu s$

$$T_{bit} = \frac{1}{bitrate}$$

EX.03 – Example of Calculation Bit Time

For bitrate = 500 kbit/s

$$T_{bit} = \frac{1}{500 \times 10^3} = 2\mu\text{s}$$

For bitrate = $1Mbit/s$

$$T_{bit} = \frac{1}{1 \times 10^6} = 1\mu\text{s}$$

2.2.3 Propagation Time T_{prop} – Derivation

Propagation delay is the time it takes for a signal to physically travel along a medium (e.g., copper wire).

$$T_{prop} = \frac{L}{\text{v}}$$

L: Length of the cable (meters)

v: Propagation speed of the signal (meters/second)

This formula comes from basic physics:

$$Speed = \frac{Distance}{Time} \quad ==> \quad Time = \frac{Distance}{Speed}$$

Signals do not travel at the speed of light c but at a reduced velocity v=$\alpha{\cdot}c$, where α ≈0.6−0.7for typical copper wires.

For CAN bus, commonly used v=0.66. $c = 198 \times 10^6\ m/s$

EX.04: If L=20m, then:

$$T_{prop} = \frac{20}{198 \times 10^6} \approx 0.101\mu s$$

To maintain proper CAN arbitration:

The propagation delay T_{prop} must be significantly smaller than the bit time T_{bit}

This ensures that all nodes see the same bit in time to resolve priority

$$T_{prop} < 0.1 . T_{bit} \textit{ is ideal}$$

2.3 Signal Scaling and PDU Mapping

In automotive communication systems like CAN, raw binary values must be mathematically transformed into physical (interpretable) values. This section presents the scaling and offsets process for signals transmitted in a 64-bit PDU, followed by a visual mapping of their positions within the data frame. [2]

2.3.1 Derivation of Scaling Factor (SK) and Offset (SO)

In digital communication (e.g., CAN bus), physical signals such as distances, speeds, or probabilities are encoded using a fixed number of bits. To map the encoded binary value (raw data) back into the corresponding physical quantity, a linear scaling function is applied. The goal is to linearly distribute the physical values over the available bit range. [3]

- A signal is encoded using n bits. [7]
- The signal's physical value lies in a defined range:

Physical Min ≤ Value ≤ Physical Max

- This range must be divided into discrete steps representable by the available bit width:

Number of Steps = 2^n

- But since digital counters start at 0, the number of intervals between representable values is:

Number of Intervals = $2^n - 1$

- The Scaling Factor (SK) is the size of each interval, i.e., how much physical quantity one bit step represents: [6]

$$\textbf{Scaling Factor (SK)} = \frac{\textbf{\textit{Physical Max}} - \textbf{\textit{Physical Min}}}{\mathbf{2}^{\textbf{\textit{Length}}} - \mathbf{1}}$$

- The Offset (SO) is the minimum physical value, which corresponds to a raw value of 0:

Offset (SO) = Physical Min

- To convert any binary-decoded value (Raw Value) into a real-world physical value:

Physical Value = Raw Value × SK + SO

2.3.2 Completed Signal Scaling Table

EX.05: Let’s calculate the scaling factor for Object ID:

Bit length: 4 bits

Physical range: 0 to 10 meters SK = $\frac{10-0}{2^4-1} = \frac{10}{15} \approx 0.666667$

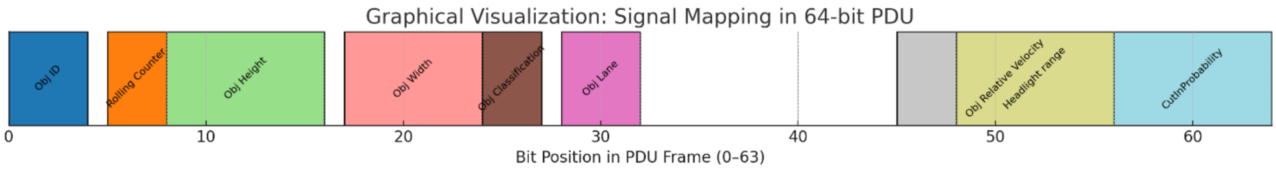
Offset (SO): 0

Note: Fields like Obj ID, Classification, or Lane are **categorical or logical** values, hence unit is not applicable.

2.3.3 Bit-Level PDU Mapping (Visualization)

To ensure consistent encoding and decoding of sensor data within automotive communication systems such as CAN, each signal must occupy a specific range of bits within the Protocol Data Unit (PDU). This section visualizes the layout of these signals in a 64-bit message frame based on the structure defined in the signal table. [6]

The following diagram illustrates the bit-level mapping of each signal:



- The horizontal axis represents the bit positions (0–63) in the PDU frame.
- Each colored bar corresponds to a unique signal (e.g., Obj ID, Obj Height, etc.).
- The length of each bar matches the bit-length defined for that signal.

Signal	Length (Bits)	Physical Range	Formula for SK	SK (Scaling Factor)	SO (Offset)	Unit
Obj ID	4	0 to 10	$\frac{10-0}{2^4-1} = \frac{10}{15}$	0.666667	0	-
Rolling Counter	3	0 to 8	$\frac{8-0}{2^3-1} = \frac{8}{7}$	1.142857	0	-
Obj Height	8	0 to 10	$\frac{10-0}{2^{10}-1} = \frac{10}{255}$	0.039216	0	m
Obj Width	7	0 to 5	$\frac{5-0}{2^7-1} = \frac{5}{127}$	0.039216	0	m
Obj Classification	3	0 to 7	$\frac{7-0}{2^3-1} = \frac{7}{7}$	1.0	0	-
Obj Lane	4	0 to 8	$\frac{8-0}{2^4-1} = \frac{8}{15}$	0.533333	0	-
Obj Relative Velocity	11	-35.0 to +35.0	$\frac{35-(-35)}{2^{11}-1} = \frac{70}{2047}$	0.034203	-35.0	m/s
Headlight Range	8	0 to 250	$\frac{250-0}{2^8-1} = \frac{250}{255}$	0.980392	0	m
CutInProbability	8	0 to 100	$\frac{100-0}{2^8-1} = \frac{100}{255}$	0.392157	0	%

- The position of each bar corresponds to the signal’s Start Bit in the message.
- Dashed vertical lines every 8 bits help delineate byte boundaries, which is especially relevant when considering byte order (Motorola format in this case).

This graphical mapping is critical in system design for several reasons:

- Ensures that each receiver ECU interprets signal values at the correct bit positions. Prevents **signal overlap** and encoding errors. Facilitates **debugging and reverse engineering** of CAN traffic using tools like Vector CANoe or PCAN-View.

2.4 Technical Properties of the OX08B40 Automotive Image Sensor

Automotive-grade image sensors like the OmniVision OX08B40 offer features tailored for ADAS and autonomous driving. Table 2-1 captures the most important specifications extracted from the datasheet and product brief: [5] [8]

Property	**Specification**
Resolution	8.3 MP (3840 × 2160 pixels), 16:9 aspect ratio
Frame Rate	Up to 36 fps at full 4K resolution
Pixel Size	2.1 µm × 2.1 µm
Optical Format	1/1.73″ diagonal
High Dynamic Range (HDR)	140 dB (dual conversion gain + onboard HDR engine)
LED Flicker Mitigation	Integrated (LFM + HALE Engine) – Low Fuel Motorsport
Signal-to-Noise Performance	Dual conversion gain with 82 dB at first exposure
Output Formats	12/14/16/20-bit raw PWL or uncompounded 24-bit combined HDR
Interface	Up to 4-lane MIPI CSI-2 - Mobile Industry Processor Interface
Power (Stream @4K 36fps)	≈743 mW (active)
Power (3-capture operation)	≈634 mW
Shutter Type	Rolling shutter
Temperature Range	−40 °C to +105 °C (sensor ambient), up to +125 °C junction
Aux Features	ASIL C compliance, embedded temperature sensor, OTP memory, cybersecurity engine

2.5 Feasibility of Transmitting Camera Sensor Data over CAN Layer 2

Modern Advanced Driver Assistance Systems (ADAS) rely heavily on data from camera sensors. These sensors can deliver information in two primary forms: [5]

1) Object Data (processed outputs such as detected object positions, sizes, velocities, etc.)
2) Raw Data (original pixel-level image data from the sensor)

In this section, we evaluate whether each form of data can feasibly be transmitted using the CAN bus at Layer 2, i.e., without higher-level transport protocols, segmentation, or compression mechanisms. [1]

- *CAN Layer 2 Constraints*
  - Maximum payload: 8 bytes (64 bits) per frame (CAN 2.0)
  - Maximum bitrate: 500 kbit/s – 1 Mbit/s
  - Net data throughput (after overhead): ~246 – 492 kbit/s (as shown in Section 1.2) [2]

2.5.1 Transmission of Object Data

Object data refers to interpreted results from the image processing pipeline, such as:

- Object ID
- Classification (e.g., pedestrian, vehicle)
- Width/height
- Lane position
- Relative velocity
- Cut-in probability

As shown in Section 2.3, all of these signals can be encoded within one to two 64-bit CAN frames using efficient scaling and offset encoding. Fully supported by CAN Layer 2. Object data is small and structured. Fits into a cyclic schedule typical of CAN-based ADAS communication. Suitable for real-time vehicle functions like braking, steering, and warning.

2.5.2 Transmission of Raw Image Data

- Not at all like protest information, crude picture information alludes to the natural pixel yield captured by the camera sensor. This incorporates person pixel force for each outline, in groups like grayscale, RGB, or Bayer-pattern. Whereas fundamental for preparing AI models and nitty gritty scene examination, crude information forces extraordinary transfer speed requirements. [5]
- This segment assesses whether it is actually attainable to transmit crude picture information from a camera sensor by means of the CAN transport at Layer 2.
- Let us consider a typical camera configuration:
- Resolution: 1280 × 720 pixels (HD)
- Bit depth: 8 bits/pixel (grayscale)
- Frame size: 1280 × 720 = **921,600 bytes ≈ 7.37 Mbit per frame**
- Frame rate: 30 fps
- Required bandwidth: 30×7.37=**221Mbit/s**
- Not feasible to transmit raw camera data over CAN Layer 2. Even one single image frame would take over 15 seconds to transmit—completely impractical for real-time applications.

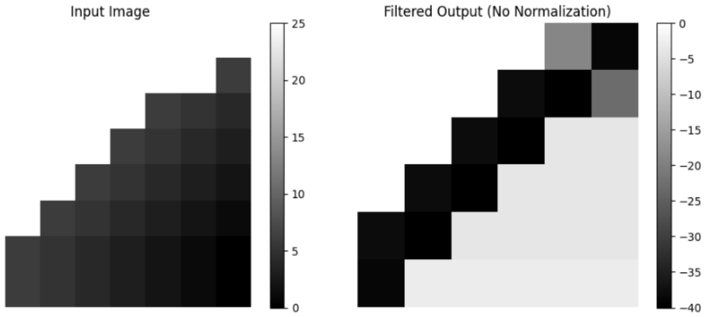
2.6 Connecting an Analysis Tool to the CAN Bus

To monitor CAN traffic, an analysis tool (e.g. Vector CANoe, PCAN-View) connects directly to the CAN_H and CAN_L lines using a USB-to-CAN interface (e.g. PEAK, Kvaser, or Vector adapters). This is typically done via: [7]

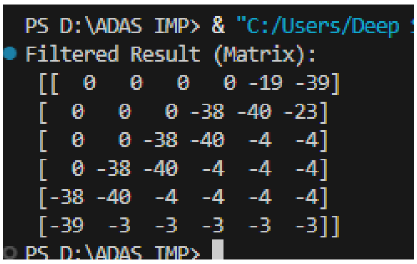
- OBD-II diagnostic port
- Breakout connector
- Test points on the wiring harness

The adapter acts as a passive listener and doesn't interfere with normal communication. Software reads all messages, applies a DBC file to decode signal content, and timestamps each frame.

2.7 Application of Edge Detection Filter to Grayscale Image



Strong negative values appear diagonally from top-right to bottom-left—indicating a diagonal intensity drop in the input. The filter detects areas where brightness (gray value) shifts abruptly across both horizontal and diagonal edges. This is a diagonal & cross-edge detection filter, designed to emphasize changes in intensity across diagonals, left–right, and up–down simultaneously. Zero-padding (fillvalue =0) was used to define values outside the image boundaries. This creates an output of smaller size (valid mode) and avoids undefined regions.



```
PS D:\ADAS IMP> & "C:/Users/Deep
Filtered Result (Matrix):
 [[  0   0   0   0 -19 -39]
 [  0   0   0 -38 -40 -23]
 [  0   0 -38 -40  -4  -4]
 [  0 -38 -40  -4  -4  -4]
 [-38 -40  -4  -4  -4  -4]
 [-39  -3  -3  -3  -3  -3]]
PS D:\ADAS IMP>
```

2.8 Literature Summary: Multi-Modal 3D Object Detection

The paper "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey and Taxonomy" by Wang et al. (2023) provides a comprehensive survey of methods that integrate multiple sensor modalities—primarily camera, LiDAR, and radar—for 3D object detection in autonomous vehicles. The authors argue that combining complementary sensor types leads to significantly improved detection accuracy, robustness, and scene understanding compared to using single-modality systems. [1] [4]

Key Contributions:

- Taxonomy Structure: The paper introduces a detailed and structured taxonomy for multi-modal detection architectures based on three stages:
  - Representation: How raw sensor data is formatted (e.g., image features, voxels, BEV maps).
  - Alignment: Techniques to register data spatially/temporally (e.g., calibration, learned transformation).
  - Fusion: Strategies to combine data (e.g., early fusion, late fusion, attention-based, transformer-based fusion).
- Fusion Mechanisms:
  - Learning-agnostic methods (e.g., concatenation, pooling) are simple but less flexible.
  - Learning-based methods (e.g., attention, transformers) offer adaptive and robust fusion, especially in occluded or sparse scenes.
- Challenges Identified:

    - Calibration and synchronization between heterogeneous sensors.
    - Real-time constraints for embedded deployment.
    - Handling rare classes or domain shift (e.g., bad weather, long-tail objects).
    - Balancing fusion depth, latency, and computational overhead.
- Future Directions:
    - Transformer-based fusion architectures.
    - Temporal and sequential data integration.
    - Lightweight models for real-time deployment.
    - Self-supervised learning and domain generalization. [5]

3. Results

The implementation and evaluation tasks carried out in this work span multiple aspects of automotive sensor analysis and communication: A convolution kernel was applied to a grayscale image to detect diagonal edges, yielding a clear contrast map showing the intensity transitions (Section 2.7). The result revealed sharp edge detection along the main diagonal where the pixel intensity rapidly dropped. Signal scaling and PDU mapping were completed for a camera-derived object detection system. Calculated scaling factors and offsets ensured physical interpretation of values like object velocity, height, width, and cut-in probability within CAN frames. These signals were successfully visualized in a 64-bit layout (Section 2.3). Distance estimation using stereo camera disparity was derived and demonstrated with realistic parameters, confirming depth estimation accuracy in the range of 8 meters based on pixel disparity (Section 2.1).

4. Discussion

The evaluation confirmed that camera-derived object data is compact and suitable for real-time ADAS communication over CAN Layer 2, while raw image data exceeds CAN's bandwidth limits, requiring Ethernet or edge processing. The applied convolution filter effectively detected diagonal edges, proving the usefulness of lightweight image processing. The OX08B40 sensor was identified as ideal for ADAS due to its high resolution, HDR, MIPI CSI-2 interface, and ASIL C compliance.

5. Summary and Outlook

This study examined the integration of camera systems in vehicles from three key perspectives: sensing, processing, and data transmission. It was shown that object data can be effectively compressed and transmitted over CAN, whereas raw image data exceeds bandwidth limits. High-resolution sensors such as the OX08B40 offer significant advantages in detection quality. Multi-modal fusion further improves accuracy but introduces computational complexity. Future ADAS systems should leverage Ethernet for high-volume data, deploy edge AI for real-time processing, and utilize advanced fusion techniques (e.g., transformers) to support scalable and intelligent vehicle perception.

6. List of references, web links and citations

Literature Sources

[1] H. Winner, S. Hakuli, F. Lotz, and C. Singer (Eds.), *Handbuch Fahrerassistenzsysteme – Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, 3rd ed., Springer Vieweg, Wiesbaden, 2015. ISBN: 978-3-658-05733-6.

[2] Prof. H. Harasim, *Autonomous Systems*, Hochschule Landshut Lecture Material, 2008.

[3] K. Reif, *Automobil Elektronik*, Springer Vieweg, Wiesbaden, 2006. ISBN: 3-528-03985-X.

[4] B. Yang, M. Liang, and R. Urtasun, "Multi-Modal 3D Object Detection in Autonomous Driving: A Survey and Taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. DOI:10.1109/TPAMI.2023.3269513

Lecture PDFs (Prof. Thomas Limbrunner, THD)

[5] 250513_ADAS_08.pdf, Hochschule Landshut Lecture Script (ADAS08).

[6] 250429_ADAS_06_1.pdf, Hochschule Landshut Lecture Script (ADAS06).

Web References

[7] Vector Informatik GmbH, *CAN Protocol & DBC File Format Documentation*, 2007. Available: https://vector.com

[8] OmniVision Technologies, "OX08B40 Image Sensor Datasheet." [Online]. Available: https://www.ovt.com/products/ox08b40/