



Advanced Driver Assistance Systems and Autonomous Driving

11

This chapter discusses advanced driver assistance systems (ADAS) and autonomous driving. ADAS are systems to help the driver in the driving process. When designed with a safe human-machine interface (HMI), they could increase vehicle safety and more in general road safety. Autonomous driving is based on increasing vehicle automation that leads to autonomous or self-driving or semiautonomous vehicles. Self-driving vehicles are one of the major drivers of change in the automotive industry. This has been discussed in Chap. 2, showing how major OEMs responded, for example, the creation of the CASE organization in Daimler, the introduction of Chief Digital Officer (CDO) positions in BMW and VW, and others. Section 11.1 builds on the introductory treatment of ADAS in Chap. 4 and gives an overview of commercial ADAS functionalities and the sensors being used. Section 11.2 gives a quick recap of the main ADAS functions like lane keeping, lane departure warning, and others in more detail. The section refers to the situation of objects moving across, either in front or behind, the vehicle and advanced methods for pedestrian and object detection.

ADASs are part of active safety initiatives and have been able to drive down the number of fatal accidents (see Chap. 2). Camera-based ADAS systems need sophisticated image processing and analysis algorithms, for example, image processing for lane keep assistance with preprocessing, edge detection, and line segmentation. Therefore, Sect. 11.3 discusses the basic principles of image processing and important algorithms for this vast topic and also shows, how MATLAB and Simulink can be used for rapid prototyping of camera-based ADAS functionality by using the Image Processing Toolbox.

Section 11.4 takes a birds-eye look at the transition from ADAS to autonomous driving summarizing the essential steps in a mindmap, while Sect. 11.5 gives a quick overview of the legal framework and liability issues for autonomous driving.

In Sect. 11.6 a typical ADAS software architecture is shown and various middleware technologies that are currently being evaluated for use in the design of autonomous cars. One of these middleware technologies is SOME/IP, which was developed by BMW and is also available as a standard AUTOSAR implementation. Autonomous cars will not only rely on onboard sensors but need information from infrastructure, maps, and other vehicles. This defines a complex attack surface for cyber threats which is analyzed in detail in the Sect. 11.7. The focus is on cyber threats and functional safety. Cybersecurity solutions like the ones discussed in the previous chapters can be deployed to secure autonomous driving. Section 11.8 wraps up with a summary, and recommended readings, while Sect. 11.9 contains a comprehensive set of questions on advanced driver assistance systems and autonomous driving. Finally, the last section includes references and suggestions for further reading.

11.1 Advanced Driver Assistance Systems

Active and passive safety is an area of intense research and continues to be a field where automakers can differentiate themselves from each other. Active safety includes brake assistance, traction control, electronic stability programs, and ADAS, while passive safety systems include seat belts, airbags, crashworthiness, and so forth. Active safety systems reduce the probability of accidents and sometimes even avoid accidents altogether, while passive safety systems help to reduce the impact on the human passengers. With rapid advances in the field of sensors, mechatronics, and computer vision, driver assistance functionalities like:

- Adaptive cruise control (ACC)
- Speed limit assistance
- Blind spot detection (BSD)
- Driver monitoring and drowsiness detection systems
- Emergency brake assistance
- Intelligent headlamp control
- Intelligent parking assistance
- Lane departure warning
- Night vision
- Obstacle and pedestrian detection
- Traffic sign recognition, etc.

have become affordable even in the midrange segment of cars.

Chapter 11 discusses parking assistance systems as one of the popular and widely deployed driver assistance functions. Figures 11.1 and 11.2 illustrate such a ParkPilot functionality in a modern car, a current version of the Seat Leon. Sensors monitor the front and back drive path and alert the driver if the car comes too close to an obstacle. This is a very helpful feature as visibility is often constrained, and many obstacles are not clearly visible from the driver's seat.



Fig. 11.1 ParkPilot ADAS function in a Seat Leon—front view



Fig. 11.2 ParkPilot ADAS function in a Seat Leon —side and rear view

Advanced driver assistance systems cover a wide range of application scenarios. The mindmap of Fig. 11.4 differentiates between vehicle support and driver support systems. The latter systems provide valuable information to the driver, improve perception, and detect critical conditions that affect the driver's performance. Typical systems in this category of ADAS are (see also Chaps. 4 and 5):

- *In-vehicle navigation system*: These systems guide the driver visually and acoustically (text-to-speech) based on built-in maps and sophisticated route planning. A satellite navigation system provides autonomous geo-spatial positioning with global coverage. It allows small electronic receivers to determine their location (longitude, latitude, and altitude) to within a few meters using time signals transmitted along a line-of-sight by radio from satellites.
- *Drowsiness detection*: Monitors the driver by evaluating different parameters, e.g., steering movements and alerts if a critical level of tiredness has been reached. A tired driver might doze off for a few seconds which can lead to severe accidents on the highway.
- *Automotive night vision*: It is a system to increase a vehicle driver's perception and seeing distance in darkness or poor weather beyond the reach of the vehicle's headlights. Night vision systems are currently offered as optional equipment on certain premium vehicles.

Vehicle support systems include:

- *Adaptive cruise control (ACC)*: Adaptive cruise control (also called radar cruise control) is an optional cruise control system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead. In its basic configuration, it makes no use of satellite or roadside infrastructures or of any cooperative support from other vehicles; the control algorithm relies on sensor information from onboard sensors only. The extension to cooperative cruise control requires either fixed infrastructure as with satellites, roadside beacons, or mobile infrastructures as reflectors or transmitters on the back of other vehicles ahead. These systems use either a radar or laser sensor setup allowing the vehicle to slow down when approaching another vehicle ahead and accelerate again to the preset speed when traffic permits.
- *Lane departure warning (LDW), lane keep assistance (LKA), and Lane change assistance (LCA)*: An LDW system is designed to warn a driver when the vehicle begins to move out of its lane (unless a turn signal indicates the wish to leave the lane) on freeways and arterial roads. These systems are designed to minimize accidents by addressing the main causes of collisions: driver error, distractions, and drowsiness. Lane keep assistance systems actively control the lateral movements of the car to stay in a given lane. The next step in higher automation is lane change assistance (LCA). A LCA system can not only keep a lane but change it automatically when the driver sets the turn signal indicator. This requires an active monitoring of the traffic approaching from behind. The system will look for a safe gap in the traffic flow and automatically change the lane when possible by controlling the steering wheel.

- *Collision avoidance (pre-crash) system (CAS)*: Is an automobile safety system designed to avoid accidents or at least reduce the severity of an accident. Also known as a pre-crash system, forward collision warning system or collision mitigating system, radar, laser, and camera sensors are used to detect an imminent crash.
- *Automatic parking (AP)*: This is an autonomous car-maneuvering system that steers the car from a traffic lane into a parking place to perform parallel parking, perpendicular parking, or angle parking. These ADAS functions enhance the comfort and safety of driving in constrained environments, where a lot of care and experience is required to steer the car. The topic was discussed in detail in Chap. 10.

The market for ADAS systems is growing substantially, and the margins are still relatively high.

The market is dominated by:

- Automotive OEMs
- Device manufacturers and suppliers (1st Tier, 2nd Tier. . .)
- Start-ups focusing on special aspects like sensors, artificial intelligence, or frugal implementation of ADAS features
- Semiconductor companies

The device manufactures provide a multitude of components like radar and lidar sensors, camera systems, infrared sensors, alarm systems, illumination and sound alert systems, display units, and others.

Most automotive OEMs offer some ADAS functionality in their cars either optional or as standard. Premium OEMs like BMW, Audi, Daimler, etc. provide sophisticated ADAS features.

Major suppliers in the ADAS space are Continental, Bosch, Delphi, ZF/TRW, Visteon, Mobileye (acquired by Intel), and Valeo ([URL9 2017](#); [URL10 2017](#); [URL1 2017](#)).

Semiconductor companies are also active and moving up the value chain, among them NXP/Freescale Semiconductors (ICs), Infineon Technologies (ICs), and Intel which strengthened its position with the takeover of the Israeli ADAS specialist Mobileye.

Figure 11.3 illustrates the main sensor categories that are being used for advanced driver assistance functions:

- Ultrasound
- Camera/video
- Camera/infrared
- Radar (near and long distance)
- Lidar

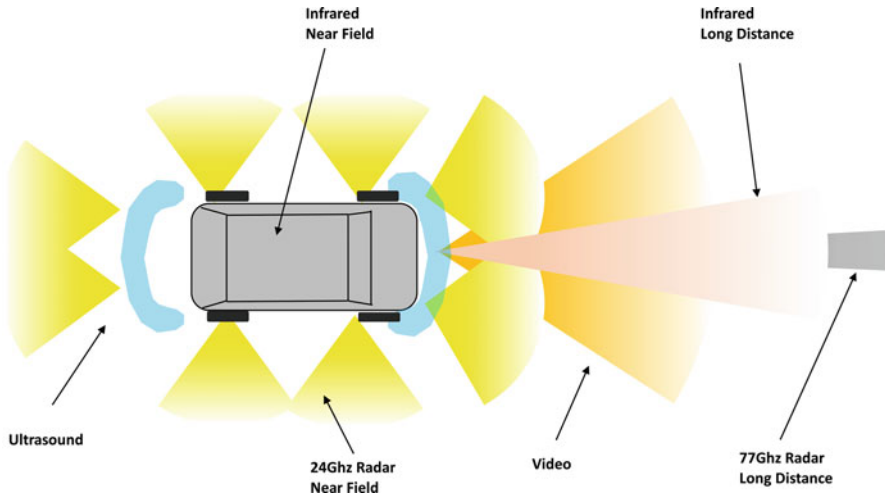


Fig. 11.3 Sensor application for different ADAS functions (see URL22 2017)

These sensors differ in resolution, range, accuracy, and price. With rapid advances in hardware and software, the prices for ADAS functions are coming down significantly.

Sensors were also discussed in Chap. 4, Sects. 4.2.4 and 4.2.7, and ADAS functions in Chap. 5, Sects. 5.4.1 and 5.5. The discussion in these sections is built on this material.

Figure 11.4 shows a mindmap (Müller and Haas 2014) which groups the different ADAS functions into two main categories, vehicle support and driver support.

11.2 Lane Departure Warning, Lane Keep Assistance, Obstacle Detection, and Crossing Assistance

11.2.1 Lane Keeping and Lane Change Assistance

The basic lane detection approach was described in Chaps. 4 and 5. In Fig. 11.5 the lane keep assistance function in a Seat Leon is shown. As per the current law, the system is not allowed to take over full control but requires the driver to touch and move the steering wheel every 30 s or so, as shown in Fig. 11.6. If the driver fails to do so, the car will send a warning, both visually (see Fig. 11.6) and acoustically and ultimately slow down and stop if no activity of the driver is being detected.

There are two main types of lane assistance systems:

- Systems which warn the driver (lane departure warning system, LDW) if the vehicle is leaving its lane (visual, audio, and/or vibration warnings)
- Systems which warn the driver and, if no action is taken, automatically take steps to ensure the vehicle stays in its lane (lane keeping assistant, LKA)

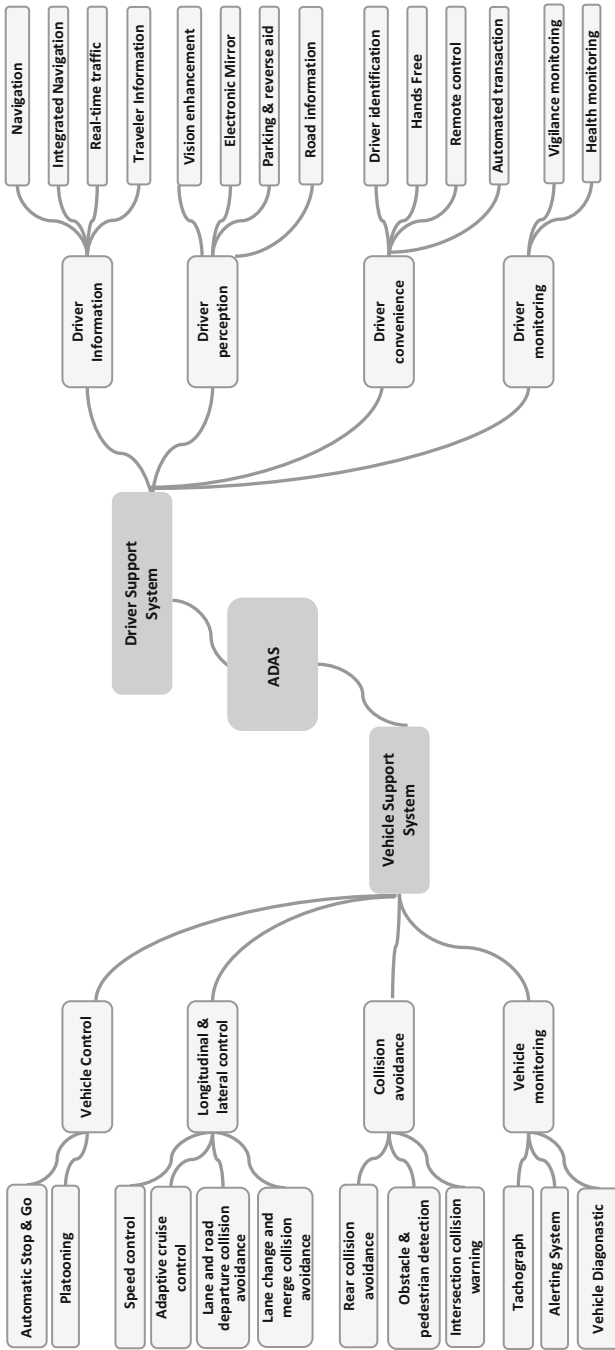


Fig. 11.4 Mindmap of ADAS functions (Müller and Haas 2014)



Fig. 11.5 Lane keeping assistant (LKA) ADAS function in a Seat Leon



Fig. 11.6 Lane keeping assistant (LKA) ADAS function in a Seat Leon; driver must take up control again

LDW can help prevent single vehicle roadway departure, lane change or merge collisions, and rollover crashes, as described below:

- *Single vehicle roadway departure:* LDW gives a warning as the car crosses the shoulder lane marking. Without the system, the car may be driven off the shoulder and crash into off-road obstacles, for example, light poles, signs, guardrails, trees, and stopped vehicles.
- *Lane change/merge:* LDW issues a warning as the car crosses center lane markings on multilane roadways, including solid lines, double lines, dotted lines, dashed lines, and raised pavement markers. Without the system, the car may be driven into an adjacent lane, resulting in a head-on or sideswipe collision.
- *Rollovers:* LDW may also prevent some crashes that would be categorized as rollover crashes. For example, if the vehicle drifts out of the lane onto the shoulder, the car could roll over if a sudden recovery maneuver is made.

LDW may also:

- Assist the driver in consistently keeping a vehicle in the lane, thereby reducing lane departure crashes.
- Reinforce driver awareness of vehicle position in the lane to maintain a more central lane position and improve the driver's attentiveness to the driving task.

LDW cannot prevent all single vehicle roadway departure crashes. These are warning devices and do not actively prevent crashes, they warn the driver so he/she can maneuver the car to prevent a crash. For example, crashes involving vehicle loss of control due to slippery roads and excessive speed on turns would not be prevented with these systems. Also, the systems will not prevent crashes due to intentional lane changes, which involve the driver's failure to see another vehicle in the adjacent lane or by a vehicle being in a blind spot. Some collision warning systems (CWS) have blind spot sensors to help prevent these types of crashes.

LDW should work under various operational scenarios:

- *Normal system start-up operation:* When the driver turns the ignition switch to start the vehicle, the LDW performs a power-up self-test, and the driver scans the warning indicator to determine any system malfunctions. If necessary, the driver may alert fleet maintenance for corrective action. When the vehicle reaches the minimum LDW tracking speed on a roadway with lane boundary markings, lane tracking begins.
- *Warning/alert situations:* When travelling at or above the minimum LDW tracking speed, a driver may unintentionally drift out of the lane. Then, the LDW issues a warning.
- *System fault conditions:* When the LDW cannot track the lane or a system fault occurs, the driver is notified via the lane-tracking indicator. This inability to track lanes may be due to a lack of lane markings, poor quality of lane markings, poor

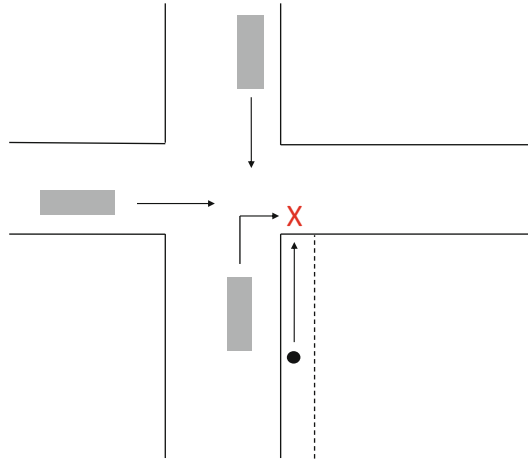
visibility, or a dirty/icy windshield. Although LDW cameras typically view the road through a portion of the windshield swept by the wipers, the driver can manually clean the windshield area in front of the LDW camera to see if the LDW begins to track. Some LDW may display various messages when certain types of faults or other conditions are detected, such as *Calibration in Progress*.

- *Well-marked roads*: The most commonly encountered roadway markings include single and double solid lines, dashed and dotted lines, and raised pavement markers where LDW should detect lane departures and issue warnings to a driver travelling over the minimum tracking speed.
- *Roads with missing or degraded lane boundary markers*: If lanes have missing or degraded lane markings, the driver may not receive a warning as the vehicle progresses outside of the lane, depending on the particular LDW used. On roads with only one set of markers, the driver should receive a warning when the warning threshold is crossed on that side, even if the system cannot detect the lane boundary on the other side.
- *Delivery points, arterials, and collectors*: Currently available LDW systems will not operate at delivery points and roads where the car travels at speeds below the minimum LDW tracking speed. Currently available LDW systems are made primarily for highway driving and will not function at lower speeds associated with some local roads.
- *Wet roads*: Due to reflections on wet road surfaces, LDW may occasionally be unable to detect lane markings; however, the lane-tracking indicator will show that the system is not providing warnings under these conditions.
- *Mud-/ice-/snow-covered roads*: When lane markings are not visible on roads covered by mud, ice, or snow, the lane-tracking indicator will show that the system is inactive. LDW may be beneficial in low visibility conditions (e.g., rain, fog, and falling snow) when lane markings are present.

As outlined in the previous section, lane keep assistance systems actively control the lateral movements of the car, thereby making sure that the car stays in the lane. Lane change assistance systems can perform a lane change maneuver on their own if the driver sets the turn signal indicator. This is a complex process which involves sophisticated image and sensor processing:

- *Analysis of traffic approaching from behind*: Vehicles with different speeds are approaching and need to be detected. The velocity needs to be estimated.
- *Space detection*: The system has to look for a safe gap in the traffic flow, based on the relatively velocity of vehicles approaching.
- *Maneuver control*: If an appropriate gap in the traffic flow is detected, the system will automatically change the lane by controlling the steering wheel.
- *Switch back to LKA mode*: Control is handed over to lane keep assistance.

Fig. 11.7 Turn assistance, avoiding collisions with pedestrians and cyclists



11.2.2 Turn Assistance

A serious course of accidents is the collision with pedestrians or cyclists in busy city traffic (URL13 2015; URL30 2017). Especially, when turning into a side street, the driver has to look for crossing pedestrians or cyclists. Unfortunately, it is quite common that a cyclist, pedestrian, or animal is overlooked approaching from the side path.

Turn warning systems, turning assistance systems, and obstacle detection systems can identify these dangerous situations and help the driver to prevent accidents which often have severe consequences (see Fig. 11.7).

If an object is moving across, either in front or behind the vehicle, it is relatively easy to detect this. Also, if an object moves slowly towards a vehicle, the ADAS image analysis system has no problem to recognize this as the object will appear bigger when it approaches the vehicle. But the problem arises when pedestrians or cyclists are present in the blind spot. To avoid this situation, surround view systems give a complete view around the car.

Although, intersections represents a relatively small portion of a cyclist's travel route, these areas are where a cyclist is most at risk of getting hit by a car (URL29 2017).

Another possible challenge arises when the cyclist is moving very fast and intersects with the car's path. In this case, one can deduce that the cyclist is moving fast from optical flow but predicting whether he will intersect with the car's path or not is difficult. The key is to estimate the relative velocity of the cyclist as accurately as possible.

11.2.2.1 Pedestrian Detection and Object Detection Challenges

Pedestrians are often difficult to detect due to the following:

- Various styles of clothing
- Presence of occluding accessories

- Frequent occlusion between pedestrians
- Shadows
- Poor light conditions
- Different speeds
- Different shapes

11.2.2.2 Existing Approaches to Object/Pedestrian Detection

Object detection and tracking algorithms can be classified into object-based and non-object-based depending on object properties such as feature, model, and speed. The object-based approach analyzes the video on a frame-by-frame basis and detects the object based on its shape and motion. When no information is available about shape or motion, it is much more difficult to detect objects. In this case object detection is done based on previously calculated data such as other object information, location, time, and environment. These types of objects are known as indistinguishable objects, and the method is known as non-object-based detection.

The part-based model represents the body/object as deformable configurations of individual parts which are in turn modeled separately in a recursive manner. One way to visualize the model is a configuration of body/object parts interconnected by springs. The spring-like connections allow for the variations in relative positions of parts with respect to each other.

Another approach is to model pedestrians as collections of parts. Part hypotheses are firstly generated by learning local features, which include edge features, the orientation features, etc. These part hypotheses are then joined to form the best assembly of existing pedestrian hypotheses. This approach is attractive, but part detection itself is a difficult task.

A straightforward implementation is capturing and resizing the image (cropping the image inside a bounding box) to a fixed height and width. Features are extracted from these resized regions. The obtained features are then clustered to get the root filters. Given a root filter, $k \times d$ part filters are initialized at twice the spatial resolution in order to capture part details more precisely. Individual part locations are selected in two stages.

A further approach is the greedy initialization. The obtained part filters are greedily matched to image regions in order to maximize the energy map. The energy map is the squared norm of positive filter weights in each filter cell. The image regions which are matched are not considered later for matching other parts.

Ultimately, the re-defilement approach is applied using a stochastic search. After all the parts are matched, these are displaced one at a time randomly to maximize the amount of energy covered. Displacement results in a penalty which is proportional to the magnitude of the displacement. When no more energy can be covered, this phase is restarted. This process is repeated many times to avoid the selection of local maxima (Bhattacharjee 2013).

11.3 Image Processing and Image Analysis

ADAS functions depend on the combined evaluation of different sensors, like:

- Camera
- Infrared
- Lidar
- Radar
- Ultrasound

This section shows how image processing and image analysis can be applied to implement ADAS functions.

11.3.1 Computer Vision and Machine Vision

Computer vision and machine vision are terms that are being used synonymously (Davies 2012). The subjects deal with the analysis, design, and implementation of algorithms, hardware, and software for handling complex vision problems that humans or animals deal with every day. Machine vision has evolved rapidly and has benefited from the explosive growth in semiconductor performance and computer architecture.

There are two main applications for machine vision: navigation and manipulation.

- Navigation is the process of moving from one position to another.
- Manipulation is the process of actively handling objects, for example, by means of a robot manipulator.

Navigating in an unstructured, unknown environment is a complex task which is being mastered by humans and animals in a seemingly effortless manner. For a robot this is a challenge, as it has to maneuver through the environment avoiding obstacles, walls, stairs, pits, and others. One of the authors worked in a research group with researchers from Carnegie Mellon and Stanford university at the Daimler-Benz Technology Center in Berlin which developed a mobile robot in the 1990s. The computational complexity was so high that much of the calculations had to be done on a separate server, outside the robot platform, sending the results back to the robot.

Today, even smartphones have enough computational capabilities to run complex image processing and analysis programs, and indeed, some frugal ADAS applications use the smartphone's cameras and processing capabilities for obstacle detection, traffic sign recognition, and lane departure warning.

The subject of machine vision still offers a lot of unsolved research problems. Interdisciplinary by nature and being part of the vast field of artificial intelligence, machine vision is currently one of the busiest and most interesting research domains.

Computer graphics deals with the virtual construction of scenes on the screen from geometric information and description. Computer vision can be regarded as the inverse process of interpreting an image which is being captured by sensors like a stereo camera.

Often, complex image analysis problems can be broken down into simpler tasks and simplified by assumptions and knowledge about the physical scene. This is an important aspect if it comes to implementation of ADAS functions like lane keep assistance, which need to identify markings on the road.

Therefore, Sect. 11.3.2 discusses some basic principles of image processing like digital images, color models, conversion from one color space to another, spatial filtering, edge detection, and thresholding. Also, the section briefly presents the concept of morphological operations. Section 11.3.3 gives an overview of the detection of moving objects with sub sections like object tracking algorithms, and others. Section 11.3.4 introduces the optical flow algorithm, and Sect. 11.3.5 explains the implementation of image processing algorithms in MATLAB.

11.3.2 Basic Principles of Image Processing

11.3.2.1 Digital Images

An image can be considered to be a function from \mathbb{R}^2 to \mathbb{R} , mapping every point in the 2D plane to a particular real value of light and color perception. These values can be grayscales, for example, from 1 to 255, normalized intensity levels in the interval $[0,1]$, 0 and 1 for pure black-and-white (b/w) pictures or color vectors. If the values are normalized grayscales, the image $G(x,y)$ is described by the mapping

$$G : \mathbb{R}^2 \mapsto \mathbb{R}$$

$$(x, y) \mapsto G(x, y) \in [0, 1].$$

A single point on the plane and its corresponding value is called a *pixel*.

The digital image is being derived by quantizing the 2D space and mapping the real color or intensity values to discrete values. The image is characterized by its spatial resolution, the color model, the resolution, and the number scheme for the color or intensity values, for example, 8 bit values for red, green, and blue color. The spatial resolution is defined by the sensor, e.g., the elements of a charge-coupled device (CCD) sensor. There is a delicate trade-off between resolution, size of the sensor pixel element, cost, noise, and sensitivity to light.

A black-and-white image $G_{BW}(x,y)$ with a spatial resolution of 1024 x 1024 is a matrix G_{BW}

$$(g_{ij}) \quad i = 1, \dots, 1024; \quad j = 1, \dots, 1024; \quad g_{ij} \in \{0, 1\}.$$

An image consisting of intensity information will be denoted by $I(x,y)$.

11.3.2.2 Color Models

There are different color models like (Gonzalez et al. 2008):

- RGB – represents the red, green, and blue component of the image.
- NTSC – refers to the National Television System Committee standard used for color and monochrome television sets. Images are represented by three components: luminance (Y), hue (I), and saturation (Q).
- YCbCr – The YCbCr color space is used extensively in digital video. In this format, luminance information is represented by a single component, Y, and color information is stored as two color-difference components, Cb and Cr. Component Cb is the difference between the blue component and a reference value, and component Cr is the difference between the red component and a reference value.
- CMY – Cyan, magenta, and yellow are the secondary colors of light.
- CMYK – based on the CMY color model adding black as a fourth color for creating true black which is the predominant color in printing.
- HSV – refers to hue, saturation, and value. It is one of the several color systems used by people to select colors from a palette.
- HSI – refers to hue, saturation, and intensity. Decouples the intensity component from the color – carrying information (hue and saturation) in a color image.

RGB is a simple additive model where the colors are being generated by mixing three channels of basic colors that can generate any desired color. This is well suited to computer graphics, but humans tend to describe color in a different way, by hue, saturation, and brightness. The *HSI* model provides a natural way to describe colors as a 3-tuple (H,S,I) , where H is a description of the color tone, the hue value, S refers to the saturation level, and I denotes the value of the intensity/brightness. It is easy to derive the $I(x,y)$ intensity image from the *HSI* color model as the last position in the tuple corresponds directly to the I value.

The different color models can be converted into each other. It will be shown how this is done for *RGB* to *HSI* as this is often necessary for further processing of the camera images. Note, that the I value in the *HSI* model directly responds to the intensity level of an intensity image $I(x,y)$. Similarly, an *HSI* color model can be converted into the corresponding *RGB* model.

A color in the *HSI* model can be described as vector in the color unity circle (Gonzalez et al. 2008). This vector has the angle Θ and the length S .

Let (R,G,B) be a *RGB* color value. The corresponding *HSI* values can be computed in the following steps. Firstly, the H value depends on the relation between the B and G values:

$$H = \begin{cases} \Theta & \text{if } B < G \\ 360 - \Theta & \text{if } B > G \end{cases}$$

The value Θ is computed directly from the *RGB* values as follows:

$$\Theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right)$$

The saturation level is

$$S = 1 - \frac{3}{(R + G + B)} \min(R, G, B).$$

The intensity level is given by the arithmetic mean of the *RGB* values

$$I = \frac{1}{3}(R + G + B).$$

11.3.2.3 Spatial Filters

Often, images need to undergo a sequence of preprocessing steps, e.g., to remove unwanted artifacts and noise. This can be achieved with filters in the spatial or frequency domain. A spatial filter operates on the neighborhood of a particular pixel and replaces the pixel with a function of the neighboring pixels, for example, a weighted sum of these pixels. Often, a neighborhood of one pixel in all directions is chosen.

Let W be the 3×3 neighborhood matrix around a point (x, y) :

$$W = \begin{bmatrix} g(x-1, y-1) & g(x-1, y) & g(x-1, y+1) \\ g(x, y-1) & g(x, y) & g(x, y+1) \\ g(x+1, y-1) & g(x+1, y) & g(x+1, y+1) \end{bmatrix}$$

If one replaces every pixel with the sum

$$g_0(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 g(x+j, y+i)$$

noisy peaks will be averaged, and the picture will look smoother.

Figures 11.8 and 11.9 show the result of spatial filtering for removing salt-and-pepper noise (Gonzalez et al. 2008) which was added to the original image in Fig. 11.8 and then removed with a median filter with a 3×3 window (one pixel each to the left, right, top, and bottom). The result is shown in Fig. 11.9.

11.3.2.4 Canny Edge Detection Technique

Edge detection is one of the major steps in image processing. Edges define the boundary region of an image. A good edge detection technique maximizes the probability of detecting real edges and minimizes the detection of false edges. The Canny edge detection (CED) technique is known for 30 years, and still very popular. It was created by John Canny in the 1980s (Davies 2012).



Fig. 11.8 Noisy picture of a road (Interstate I5 in California near Bakersfield) distorted by salt-and-pepper noise



Fig. 11.9 Picture after noise cancellation with a spatial median filter

Edges are detected as spikes in the intensity level of the picture. This corresponds to a derivative of the intensity level.

As a preparation for discussing the Canny edge detection technique, the approximation of derivatives in discrete images has to be explained briefly.

Let G be a discrete image with intensity/gray levels $G(i,j)$. We are interested in the discrete approximation of the gradients in the x and y direction.

Let $g(j,j)$ be an arbitrary pixel in the image G . The 3×3 neighborhood around this pixel can be arranged in the form of the matrix

$$Z = \begin{pmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{pmatrix}$$

Here z_k , $k = 1, \dots, 9$ denotes the values of the neighboring pixels, and z_5 is the value of the center pixel.

The partial derivatives G_x and G_y can be approximated by differences, using approximation methods like Sobel, Prewitt, or Roberts. With the Sobel method, one would get the following approximations:

$$G_x = (z_7 + z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

The Canny edge detection algorithm works as follows:

- First, to remove small noise, image smoothing is done by using a Gaussian filter with a specified standard deviation σ .
- An edge in the image may point towards different directions; the Canny edge detection technique uses four filters to detect horizontal, vertical, and two diagonal edges in an image. The length of the edge gradient is

$$G = \sqrt{G_x^2 + G_y^2}$$

and the direction is represented by

$$\Theta = \tan^{-1} \left[\frac{G_x}{G_y} \right]$$

where G_x and G_y are the gradients in the x and y directions. As the image consists of discrete pixels, the gradient has to be approximated by one of the methods described before.

- Next, an image thresholding operation is performed. If the value of the magnitude image is less than the predefined threshold, then it is set to zero.
- Non-maximum suppression is done to reduce the edge breadth.

- Based on the result of the last stage for detecting edges and non-edges, two thresholds T_1 and T_2 are chosen, where $T_1 < T_2$. Pixel values greater than T_2 are defined as edges, and pixels with values less than T_1 are defined as the non-edge region. Pixel values in between T_1 and T_2 are considered to be edges if they are connected with an edge pixel.
- Finally, edge linking is performed.

Lines can be detected with the Hough transform, which takes a model of the line, maps it to the discrete set of points, and then computes the distances to this model line.

11.3.2.5 Image Thresholding

Image thresholding is used for removing unwanted information and noise. There are various approaches:

- Histogram based, i.e., any change in the histogram is analyzed.
- Clustering based, i.e., gray level samples are classified into two regions—background and foreground.
- Entropy based (thresholding is done based on entropy normalization).
- Spatial based, where thresholding is performed based on pixel correlation information.
- Object attribute based, i.e., thresholding is done based on similarity.

11.3.2.6 Morphological Operation

Dilation and erosion are fundamental to morphological image processing (Gonzalez et al. 2008). Dilation is an operation that “grows” or “thickens” objects in a binary image, and erosion thins or shrinks objects in a binary image. The thickening or shrinking process is determined by a structural element (Gonzalez et al. 2008). A translation of the set A by a point b is denoted as

$$A_b = \{c | c = a + b, a \in A\}$$

The dilation of A by the set B is defined as (URL27 2017)

$$A \oplus B = \bigcup_{b \in B} A_b$$

The dilation operator is commutative and associative, i.e.,

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a$$

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

The set B is also called a structuring element that is used to complete/fill structures in the set A , e.g., broken lines, holes, etc. Computationally, the center of

the structuring element will be moved across all the pixel positions of set A . Wherever a 1-pixel overlaps a 0-pixel, this will be turned into a 1.

The erosion of the set A by another set B is (URL28 2017) described as

$$A \ominus B = \bigcap_{b \in B} A_{-b}$$

It also is commutative and associative. The erosion thins or shrinks the image in the sense that only those pixels which are completely covered by the center of the structuring element will be set to 1, and all other pixels are cleared and set to 0.

Example: Three pixels $p_1, p_2,$ and p_3 in a straight line form a structuring element, the set B , which can be arranged in the matrix

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Now let us look at the set

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The result of the erosion operation is

$$A \ominus B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The block of ones has been thinned to a line of single pixels. Mixing dilation and erosion yields the following relationships (Davies 2012):

$$A \oplus B \ominus B \neq A$$

$$A \oplus B \ominus B \subseteq A$$

$$A \ominus B \oplus B \supseteq A$$

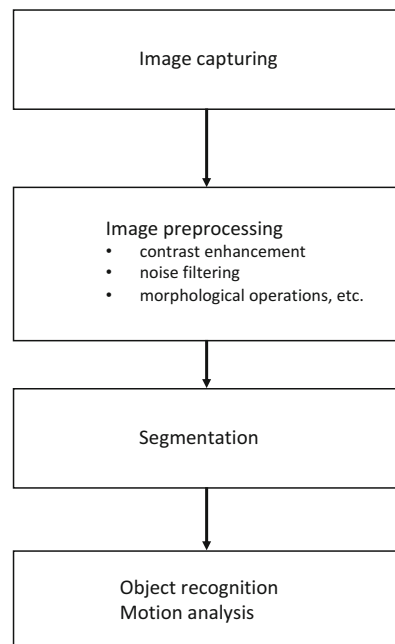
Figure 11.10 summarizes the different steps of image processing and analysis. After capturing, the image has to be preprocessed (enhancements, noise filtering, etc.). Then regions and segments have to be identified like lines (see in Fig. 11.11). Finally, the result of segmentation can be used to recognize objects. If the objects are moving, a motion analysis can be performed to find out in which direction they are moving. Object recognition is a complex process that often depends on extracting invariant features from the scene. In the next section, we will take a closer look at motion.

Figure 11.11 shows a typical road scene somewhere in the USA. The individual RGB image frames are converted into grayscale images. After that Canny edge detection is performed. Proper thresholding should be done to remove the unwanted regions. The final step uses a Hough line detection technique to extract the road markings which appear as lines (Bhattacharjee 2013). These lines and various parameters from the car like speed, steering angle, etc. could be the input for a lateral control algorithm that keeps the car in the lane.

11.3.3 Detection of Moving Objects

Object detection is an active field of research in computer vision (Rich and Knight 1991; Gonzalez and Woods 2008; Davies 2012; Haykin 2009). In this section, we

Fig. 11.10 Image processing and analysis



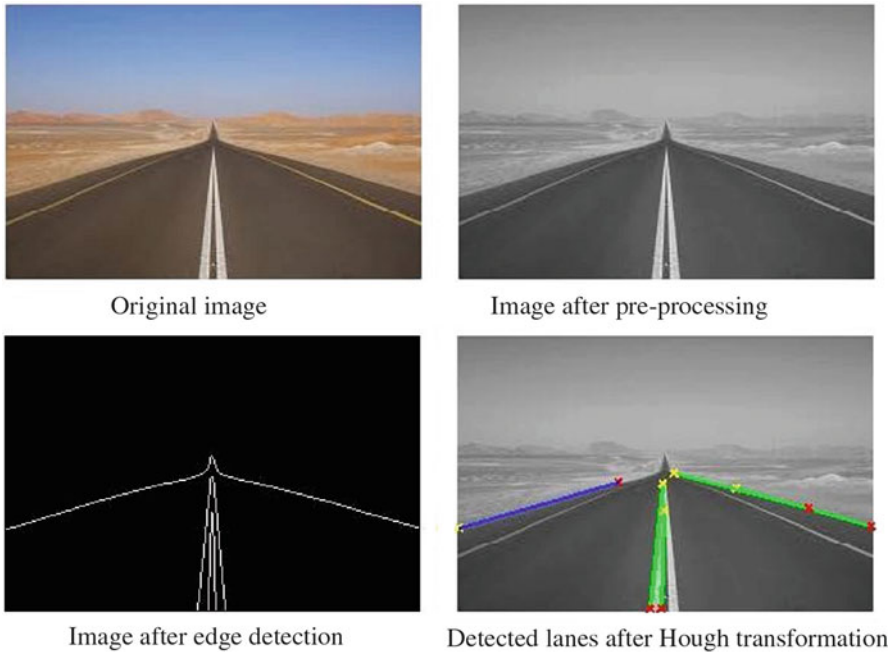


Fig. 11.11 Image processing for lane keeping assistance - preprocessing, edge detection, and line segmentation

present an approach for detecting moving objects and the direction of movement based on the analysis of Bhattacharjee (2013). The algorithms perform well under different ambient conditions like rain, fog, shadow, etc. and can detect a specific object based on the distance.

The object detection methods presented here are based on several, well-known image processing techniques and algorithms like edge detection, color space conversion, and morphological operations (Joshi 2009; Jain 2000; Gonzalez and Woods 2008). The following section gives a brief overview.

11.3.3.1 Object Tracking Algorithms

Several object tracking algorithms have been developed. Some of them are listed below:

1. Mean-shift tracking algorithm
2. Optical flow algorithm
3. Background subtraction algorithm

Mean-Shift Tracking Algorithm

The concept of the mean-shift tracking algorithm is based on the probabilistic histogram. It assumes that the object in the next frame will be located somewhere

in the vicinity of its location in the current frame. The algorithm is based on the concept of brute force tracking. The tracking method is described below:

- At first a window is defined and the target histogram is obtained.
- From the very next frame, the object tracking method begins by identifying those locations where the histogram distribution is most similar compared to the target histogram.
- For each frame, the same procedure repeats.

Optical Flow Algorithm

The optical flow can be described by the concept of relative motion between the observer and the image. Mathematically, the optical flow is a velocity field which basically wraps one image into another very similar image (Horn and Schunk 1981; Gonzalez and Woods 2008).

The literature discusses various implementations of optical flow, like Lucas-Kanade, Horn's optical flow method, Buxton-Buxton method, Black-Jepson method, general variation method, phase correlation method, and so forth (Gonzalez et al. 2008; Joshi 2009).

Background Subtraction Algorithm

Background subtraction is a method for identifying moving objects in a sequence of video frames. There are different types of background subtraction algorithms (Gonzalez and Woods 2008; Joshi 2009). In the field of computer vision, background subtraction is one of the popular methods for moving object detection because of its moderate computational effort (Gonzalez and Woods 2008). Noise can reduce the performance, and false detection is another critical issue. However, the background noise can be removed using several filters along with the main tracking algorithm; one such approach is mentioned below:

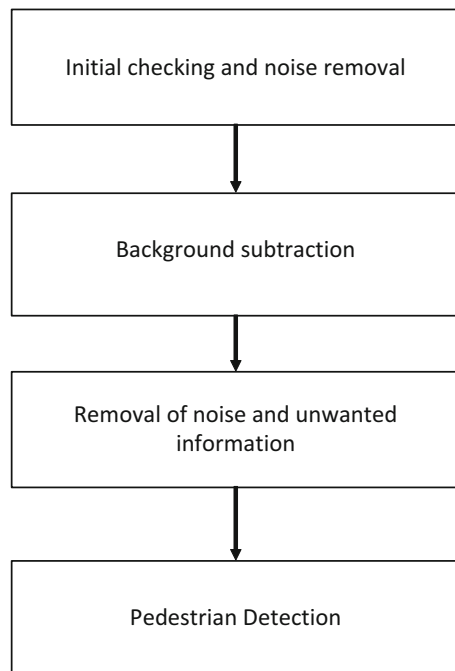
- At first the initial checkings (the aim is to check whether the camera and other related subsystems are working properly or not) need to be performed.
- After that the system checks whether anything like dust particles, leaves, paper, etc. is present in front of the camera lens (filter 1).
- The next stage deals with bad weather conditions such as rain and fog as they have a significant impact and reduce the image quality (filter 2).
- A specific problem is the presence of shadows which can lead to false detections. One way to deal with this is to remove shadows in an early stage of image processing (filter 3).
- Next background subtraction is performed to identify only those objects that are actually moving.

- Background subtraction is one of the simplest tracking algorithms. It's easy to implement but one of the major problem is information loss. One can mix several other algorithms with the actual background subtraction algorithm. For instance, an optical flow algorithm can be used to detect the direction of motion.

Figure 11.12 illustrates the concept of the detection mechanism. The following steps have to be performed:

- *Initial checking*: The purpose of the initial checking is to check whether the camera is working properly or not and to detect the presence of any occluding object on the lens of the camera. The latter condition can be recognized by computing the standard deviation of the image matrix.
- *Noise removal*: The presence of noise can degrade the performance of the system. Noise can be due to bad weather, rain, and fog. If visibility is poor, the problem can be dealt with by using histogram equalization or contrast enhancement (Davies 2012).
- *Shadow removal*: Shadows are another important aspect to consider. We are only interested in moving objects not the static objects, so the shape of the shadow changes continuously from place to place. This situation can be easily handled using the background subtraction method. After frame differencing, the shadow will appear as a small noisy portion.

Fig. 11.12 Object/pedestrian detection



By eliminating small pixel clusters, the shadow can be easily removed. Another approach is to remove the shadow at the early stages (before the background subtraction method is applied), but the problem with this approach is that, if the clothes of a person are black, then it will be difficult to distinguish between shadow (represented by a dark region) and non-shadow regions (Bhattacharjee 2013).

While tracking a car, again, the same problem may occur. The color of the car and its tires may be black and hence difficult to distinguish.

Sometimes a person may walk through a shadow region. Also, when two moving objects are very close to each other, the shadows can overlap.

There are many approaches for shadow removal (Davies 2012; Gonzalez and Woods 2008), but the problem is that most of them are condition specific; they fail to detect/remove multiple shadows of the same object. Also, the detection procedure may take a long time.

Therefore, to reduce the effect of shadows, the image frame is being converted into the *HSI* color space (RGB to *HSI*), and after that the average normalized value of each pixel is converted into the binary image. This approach is useful for minimizing the shadow effect, but it only works well on roads.

- *Background subtraction*: Using background subtraction the image foreground is extracted for further analysis. The purpose is to identify only moving objects as fast as possible. We use the frame differencing model which is defined by

$$|G(x, y, t+1) - G(x, y, t)|$$

where $G(x, y, t+1)$ is the frame at time $t+1$ and $G(x, y, t)$ is the frame at time t .

The problem with this method is that after the frame differencing operation, the effect of noise remains very high. Noise removal is done by setting a proper threshold. We use the global thresholding method (Otsu's method to minimize the intra-class variance of the black and white pixels), which calculates the value of the threshold for each frame dynamically (Gonzalez and Woods 2008).

- *Noise removal and removal of small area*: For further noise reduction, the area of each pixel cluster is calculated. At this stage, generally noisy regions (noise clusters) are fewer compared to the real object clusters (true clusters). Small noisy pixel clusters can be removed using standard cluster detection algorithms (or using a proper filter) and that way a noise free region is obtained.
- *Specific object detection*: Background subtraction and noise removal eliminates static objects and noise from the data frame. Furthermore, at this stage various cluster detection algorithms can be used to compute the size of remaining objects. One can also use a training dataset and a proper learning technique to classify these objects.

11.3.4 Optical Flow Algorithm

There are several different versions of optical flow algorithms. In this section, we discuss Horn's optical flow algorithm in detail (Horn and Schunck 1981).

11.3.4.1 Horn's Optical Flow Algorithm

Optical flow can be obtained from the relative motion between the object and the observer. The motion of the brightness patterns in the image is determined directly by the motions of corresponding points on the surface of the object.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, z + \Delta z, t).$$

Assuming that the movement is very small, and then by using Taylor series, the following approximation holds

$$I(x + \Delta x, y + \Delta y, z + \Delta z) = I(x, y, t) + \left(\frac{\partial I}{\partial x}\right)\Delta x + \left(\frac{\partial I}{\partial y}\right)\Delta y + \left(\frac{\partial I}{\partial t}\right)\Delta t + \dots$$

Combining the above equations yields

$$\left(\frac{\partial I}{\partial x}\right)\Delta x + \left(\frac{\partial I}{\partial y}\right)\Delta y + \left(\frac{\partial I}{\partial t}\right)\Delta t = 0$$

Now dividing each side by Δt gives

$$\left(\frac{\partial I}{\partial x}\right)\frac{\Delta x}{\Delta t} + \left(\frac{\partial I}{\partial y}\right)\frac{\Delta y}{\Delta t} + \left(\frac{\partial I}{\partial t}\right)\frac{\Delta t}{\Delta t} = 0$$

From this equation, the velocity components can be defined as

$$\begin{aligned}\frac{\Delta x}{\Delta t} &= u \\ \frac{\Delta y}{\Delta t} &= v\end{aligned}$$

This yields

$$I_x u + I_y v + I_t = 0.$$

The equation can be written in another way

$$(I_x, I_y) \cdot (u, v)^T = -I_t.$$

The components of the movement in the direction of the brightness gradient (I_x, I_y) equal

$$-\frac{I_t}{\sqrt{I_x^2 + I_y^2}}.$$

If neighboring points on the objects have similar velocities, then the velocity field of the brightness patterns in the image varies smoothly almost everywhere. Discontinuities in flow can be expected where one object occludes another.

Additional constraints can be expressed by minimizing the square of the magnitude of the gradient of the optical flow velocity.

One can minimize the sum of the squares of the Laplacians of the x and y component of the flow:

$$\begin{aligned}\nabla^2 u &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \\ \nabla^2 v &= \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\end{aligned}$$

Images may be sampled at intervals on a fixed grid of points. In Horn's paper (Horn and Schunck 1981), it is assumed that the measured brightness is $I_{i,j,k}$ at the intersection of the i -th row and the j -th column in the k -th image frame. Each measurement is the average over the area of a picture cell and over the length of the time interval. Therefore, the intensity values can be expressed as

$$I_x = \frac{1}{4}(I_{i,j+1,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i+1,j,k+1})$$

$$I_y = \frac{1}{4}(I_{i+1,j,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i,j+1,k} + I_{i+1,j,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i,j+1,k+1})$$

$$I_t = \frac{1}{4}(I_{i,j,k+1} - I_{i,j,k} + I_{i+1,j,k+1} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j+1,k} + I_{i+1,j+1,k+1} - I_{i+1,j+1,k})$$

By approximating the Laplacian of each u and v , we receive

$$\nabla^2 u \sim k(\bar{u}_{i,j,k} - u_{i,j,k})$$

$$\nabla^2 v \sim k(\bar{v}_{i,j,k} - v_{i,j,k})$$

where \bar{u} and \bar{v} are the local averages, one can write:

$$\begin{aligned}\bar{u}_{i,j,k} &= \frac{1}{6}(u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}) \\ &\quad + \frac{1}{12}(u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k})\end{aligned}$$

$$\bar{v}_{i,j,k} = \frac{1}{6}(v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}) \\ + \frac{1}{12}(v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k})$$

The value of the Laplacian 2D filter is

$$\begin{pmatrix} 0,5 & 1 & 0,5 \\ 1 & -6 & 1 \\ 0,5 & 1 & 0,5 \end{pmatrix}$$

By averaging these values, one gets

$$\begin{pmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{pmatrix}$$

It is important to minimize the sum of errors in the equation for the rate of change of brightness

$$\varepsilon_b^2 = I_x u + I_y v + I_t$$

The estimate of the departure from the smoothness in the velocity flow is

$$\varepsilon_c^2 = (\bar{u} - u)^2 + (\bar{v} - v)^2$$

As the image brightness measurement will be corrupted by quantization error and noise, one cannot expect ε_b^2 to be zero. This quantity will tend to have an error magnitude which is proportional to the noise in the measurement. For normalization, a weight factor α^2 (also known as regularization factor) is chosen. The total error to be minimized is

$$\varepsilon^2 = \alpha^2 \varepsilon_c^2 + \varepsilon_b^2$$

To minimize the equation, one needs to differentiate with respect to ε^2 for finding out suitable values of the optical flow velocity (u, v)

$$\frac{\partial \varepsilon^2}{\partial u} = -2\alpha^2(\bar{u} - u) + 2(I_x u + I_y v + I_t)I_x$$

$$\frac{\partial \varepsilon^2}{\partial v} = -2\alpha^2(\bar{v} - v) + 2(I_x u + I_y v + I_t)I_y$$

If one sets these two derivatives to zero one will get

$$\begin{aligned}
-2\alpha^2(\bar{u} - u) + 2(I_x u + I_y v + I_t)I_x &= 0 \\
\alpha^2(\bar{u} - u) &= (I_x u + I_y v + I_t)I_x \\
(\alpha^2 + I_x^2)u + I_x I_y v &= (\alpha^2 \bar{u} - I_x I_t)
\end{aligned}$$

Similarly, for v ,

$$I_x I_y u (\alpha^2 + I_y^2) v = (\alpha^2 \bar{v} - I_y I_t)$$

Therefore, the determinant of the coefficient matrix is

$$\begin{aligned}
(\alpha^2 + I_x^2)(\alpha^2 + I_y^2) - (I_x I_y)(I_x I_y) &= \alpha^4 + \alpha^2 I_x^2 + \alpha^2 I_y^2 + I_x^2 I_y^2 - I_x^2 I_y^2 \\
&= \alpha^4 + \alpha^2 I_x^2 + \alpha^2 I_y^2 = \alpha^2(\alpha^2 + I_x^2 + I_y^2)
\end{aligned}$$

These equations can be rewritten in the following format:

$$\begin{aligned}
(\alpha^2 + I_x^2 + I_y^2)u &= (\alpha^2 + I_y^2)\bar{u} - (I_x I_y)\bar{v} - (I_x I_t) \\
(\alpha^2 + I_x^2 + I_y^2)v &= (\alpha^2 + I_x^2)\bar{v} - (I_x I_y)\bar{u} - (I_y I_t) \\
(\alpha^2 + I_x^2 + I_y^2)u &= (\alpha^2 + I_y^2)\bar{u} - (I_x I_y)\bar{v} - (I_x I_t) \\
(\alpha^2 + I_x^2 + I_y^2)u &= (\alpha^2 + I_x^2 + I_y^2)\bar{u} - (I_x^2 u) - (I_x I_y)\bar{v} - (I_x I_t) \\
(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) &= -(I_x^2 \bar{u} - I_x I_y \bar{v} - I_x I_t) \\
(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) &= -(I_x \bar{u} + I_y \bar{v} + I_t)I_x
\end{aligned}$$

which yields

$$\begin{aligned}
(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) &= -(I_x \bar{u} + I_y \bar{v} + I_t)I_x \\
(\alpha^2 + I_x^2 + I_y^2)(v - \bar{v}) &= -(I_x \bar{u} + I_y \bar{v} + I_t)I_y
\end{aligned}$$

α^2 plays a significant role only for areas where the brightness gradient is small, preventing wrong adjustments to the estimated flow velocity due to noise in the estimated derivatives. This parameter should be roughly equal to the expected

noise in the estimation of $(I_x^2 + I_y^2)$. If one allows α^2 to be equal to zero, then one can obtain the solution of a constrained minimization problem. Applying this to the previous equation yields

$$(I_x^2 + I_y^2)(u - \bar{u}) = -(I_x\bar{u} + I_y\bar{v} + I_t)I_x$$

By solving the equation iteratively, we can write the velocity estimation equation as

$$u^{n+1} = \bar{u}^n - \frac{(I_x\bar{u} + I_y\bar{v} + I_t)I_x}{\alpha^2 + I_x^2 + I_y^2}$$

$$v^{n+1} = \bar{v}^n - \frac{(I_x\bar{u} + I_y\bar{v} + I_t)I_y}{\alpha^2 + I_x^2 + I_y^2}$$

11.3.4.2 Centroid-Based Approach

At IITB, a simple centroid based optical flow algorithm has been developed and the results are promising. Further details are mentioned below. The centroid in a plane can be computed by taking the average position of all the points in the plane. Similarly, in image processing the centroid is defined as the weighted average of image pixel intensities. It is represented by a vector that specifies the center of mass of the region. The first element represents the horizontal coordinate, while the second element represents the vertical coordinate.

If one combines the concept of centroid with the concept of Horn's algorithm, then for each different region, the centroid will be calculated, and both the concept of image centroid and Horn's optical flow depend on the image intensity. Hence, each region will be considered as a part, and the centroid of that part will be calculated, and again based on that information, the optical flow will be calculated. This integrated approach can track small movements of the object and also can detect an occluding object (Bhattacharjee 2013).

Background subtraction and the centroid-based optical flow algorithm can be used together too. Here, the background subtraction will be used to identify moving objects, and the optical flow algorithm can be used for detecting the direction of motion. In such a system, background subtraction will run as the main process, and optical flow will run as a subroutine. This approach will reduce computation time and improve the performance.

Figure 11.13 illustrates the results for various motion detection techniques analyzed by Bhattacharjee (2013).

11.3.5 Implementation Using MATLAB

MATLAB is a powerful package for numerical computation which was developed by MathWorks Inc. (URL2 2017). The popularity is due to many built-in functions



Fig. 11.13 Detection of moving objects and motion (see Bhattacharjee 2013)

for the efficient handling of matrix computations, powerful visualization, rich customization capabilities, the possibility to integrate C code, and a huge set of toolboxes for all sorts of engineering applications (Pratap 2006).

Although the focus is clearly on numerical computations, symbolic computations offered by computer algebra systems like Mathematica, Maple, and Derive are also possible with MATLAB's symbolic computation toolbox.

Together with the block-oriented simulation environment, Simulink, MATLAB offers the possibility for rapid analysis and prototyping of algorithms for a vast range of engineering applications, particularly:

- Signal processing
- System identification
- Control systems (including robust and nonlinear design methodologies)

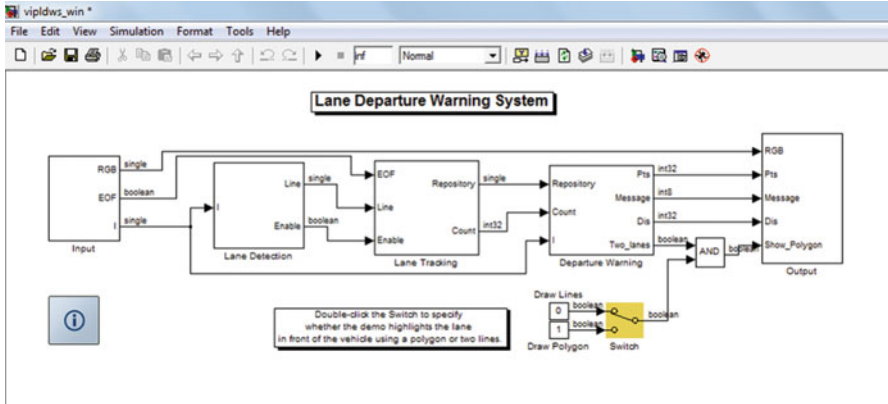


Fig. 11.14 Simulink implementation of a lane departure warning system (Gaonkar et al. 2011)

- Image processing
- Optimization problems

With the additional auto-code generation functionality (e.g., TargetLink), MATLAB programs can be directly converted to executable C and C++ code for different embedded target platforms. Figure 11.14 shows a Simulink implementation of a lane departure warning system that was implemented as a student project in the 2011 Mechatronics class at IIT-B.

11.3.5.1 Image Processing Toolbox

The Image Processing Toolbox (IPT)TM provides a comprehensive set of standard algorithms, functions, and apps for image processing, analysis, and visualization. One can perform image enhancement, image de-blurring, feature detection, noise reduction, image segmentation, and geometric transformations. Many toolbox functions are multi-threaded to take advantage of multicore and multiprocessor computers.

The Image Processing Toolbox supports a diverse set of image types, including high dynamic range, gigapixel resolution, and tomographic. Visualization functions offer the possibility to explore an image, examine a region of pixels, adjust the contrast, create contours or histograms, and manipulate regions of interest (RoIs). With the toolbox, one can restore degraded images, detect and measure features, analyze shapes and textures, and adjust color balance.

Device-independent color management enables the user to accurately represent color independently from input and output devices. This is useful when analyzing the characteristics of a device, quantitatively measuring color accuracy, or developing algorithms for several different devices.

The toolbox provides a comprehensive suite of algorithms and visualization functions for image analysis tasks such as statistical analysis, feature extraction, and object recognition.

11.3.5.2 Images in MATLAB

MATLAB's Image Processing Toolbox supports four types of images:

- RGB images
- Indexed images
- Intensity images
- Binary images

RGB images consist of three matrices, each one carrying the respective R , G , and B values. Indexed images consist of a pixel matrix and a color map. The pixel matrix contains entries (indices) into the color map. An indexed image is a matrix where every entry provides an index into an $m \times 3$ matrix of RGB values. The color map has three entries for the RGB values for each index. Grayscale images often contain 256 levels of gray levels that can be either normalized to the interval $[0,1]$ or represented using integers or byte values. A black-and-white image corresponds to a matrix of logical 0s or 1s.

The power of the IPT stems from the combination of MATLAB's optimized matrix and array operations with special image processing functions which are implemented as MATLAB m-functions. These functions provide a set of built-in operations for low-level, medium-level, and even higher-level image processing and analysis. Moreover, these functions can be even combined with algorithms from special toolboxes like neural networks and genetic optimization.

Images are loaded with the `imread` functions that can process a variety of different formats like PNG, TIFF, GIF, and JPEG. The `imshow` function displays the images on the screen, and `imwrite` stores images back on the file system.

Special MATLAB functions for low-level image processing are histogram processing, filtering, edge detection, and frequency domain transforms.

Medium-level functions are image compression and reconstruction, line, and contour detection based on the Hough transformation. There is a rich set of morphological operations and segmentation functions.

Finally, higher-level operations for object recognition are principal component analysis (PCA) and nearest neighbor operations. Many functions from other toolboxes like the neural network toolbox or the genetic optimization toolbox can be deployed to implement pattern recognition and object recognition functions.

Table 11.1 presents an overview of some of the main functions. A detailed description is given in (Gonzalez et al. 2008). MathWorks also provides details on the image processing toolbox functions on their website (URL2 2017; URL31 2017).

11.3.5.3 Statistical Functions

Statistical functions let one analyze the general characteristics of an image by:

- Computing the mean or standard deviation
- Determining the intensity values along a line segment
- Displaying an image histogram
- Plotting a profile of intensity values

Table 11.1 Some important functions of MATLAB's Image Processing Toolbox

MATLAB function	Description	Example
imread	Reads an image file from disk	<code>g = imread('anImage.jpeg')</code> reads the image <code>anImage.jpeg</code> from the file system and stores it in <code>g</code>
imwrite	Writes an image file to disk	<code>imwrite(g, 'filename.gif', 'compression', comp_value, 'resolution', res_value)</code> Stores the image <code>g</code> in the file <code>filename.gif</code> on the disk. Compression parameters and resolution is specified in the values <code>comp_value</code> and <code>res_value</code>
imshow	Displays image file on screen in a window	<code>imshow(g, 10)</code> displays an image <code>g</code> with 10 intensity levels
imhist	Computes histogram of the given image	<code>h = imhist(g)</code> <code>h</code> is the histogram of the image <code>g</code>
histeq	Histogram equalization	<code>g = histeq(f, n)</code> computes an equalized histogram of <code>f</code> with <code>n</code> intensity levels
im2bw	Converts an image into a b/w image	<code>bw = im2bw(g)</code> converts the image <code>g</code> into a black-and-white image <code>bw</code> which consist of 0s and 1s
iminfo	Shows the key information about an image	
imfilter	Linear spatial filtering	<code>g = imfilter(f, w, 'replicate')</code> computes the linear filter result of applying the filter functions <code>w</code>
fspecial	Special spatial filter	<code>g = fspecial('gaussian', [3 3], sigma)</code> Gaussian filtering of image <code>f</code> generating the resulting image <code>g</code> . The filter size is 3×3 , <code>sigma</code> denotes the standard deviation
fft2	Fast Fourier transform	<code>g = fft2(f)</code> Computes the FFT of image <code>f</code>
edge	Edge detection	<code>g = edge(f, 'canny', T, sigma)</code> computes a new image <code>g</code> that contains edges being detected by the Canny algorithm. <code>T</code> is a vector with the thresholds T_1 and T_2 , $T = [T_1 T_2]$, <code>sigma</code> is the standard deviation of the smoothing filter
imresize	Resize an image	<code>g = imresize(f)</code>

11.3.5.4 Edge Detection Algorithm

The toolbox offers a wide range of edge detection algorithms that are capable of identifying object boundaries in an image. These algorithms include the Sobel, Prewitt, Roberts, and Canny approach. The powerful Canny method can detect true weak edges without being distracted by noise.

11.3.5.5 Morphological Operators

Morphological operators enable one to detect edges, enhance contrast, remove noise, segment an image into regions, thin regions, or compute skeletons. Morphological functions in the Image Processing Toolbox (IPT) include:

- Distance transform
- Erosion and dilation
- Labeling of connected components
- Opening and closing
- Reconstruction
- Watershed segmentation

The following MATLAB code performs the morphological dilation operation. The original image in Fig. 11.15 is read from the file system, resized, converted to black and white (Fig. 11.16) and then dilated. The result is shown in Fig. 11.17.

```
g = imread('thumb_streetscene.jpg');  
g_small = imresize(g, [300 200]);  
g_bw = im2bw(g_small);g_d = imdilate(g_bw,ones(3));  
figure,imshow(g_small);  
figure,imshow(g_bw);figure,imshow(g_d);
```

11.3.5.6 Matlab Object Tracking Functions and Blocks

MATLAB provides computer vision based algorithms for object detection and tracking. They can either be used as a single function or be combined with other functions for performing complicated operations. Some of the MATLAB functions and blocks are listed below (URL32 2017):

Fig. 11.15 Image of a street in the USA after resizing



Fig. 11.16 Image converted to black and white



Fig. 11.17 Image after dilation operation



`assignDetectionToTracks`: It uses James Munkres's variant of the Hungarian assignment algorithm in the background.

`configureKalmanFilter` and `vision.KalmanFilter` class: As the name suggests, it uses Kalman filter algorithm in the background.

`vision.HistogramBasedTracker`: This function uses histogram based mean shift (CAMShift) algorithm for object tracking.

`vision.PointTracker`: This function uses the Kanade-Lucas-Tomasi (KLT) algorithm in the background.

Apart from tracking functions MATLAB also provides blocks for more complicated applications. Some of the key blocks for tracking are:

- Optical Flow
- Block Matching
- Template matching

Furthermore, some of the other MATLAB inbuilt functions can also be incorporated into an ADAS design. For instance, the `vision.ForegroundDetector` function uses Gaussian mixture models to detect the foreground. This function can be used to detect most of the moving objects in the frame. The `vision.PeopleDetector` function uses special features for people detection. This function can be deployed for pedestrian detection or crossing alert based systems.

11.4 Autonomous Driving

The topic of self-driving or autonomous cars is one of the most active areas of research (Dudenhöffer 2016; Johanning and Mildner 2015; Maurer et al. 2015; Siebenpfeiffer 2014). Most automotive OEMs and many suppliers are working in this field as we have outlined in the beginning of this chapter and throughout the book. Self-driving cars will provide mobility for the young and elderly and visually impaired, people without driver's license, and those who would like to spend travel time more efficiently. The potential benefits have triggered a race for the pole position. The complexity of the technology is enormous, and the legal framework has yet to evolve (Maurer et al. 2015). The hope is to cut down on accidents significantly as self-driving cars will have much more information available than a human driver, and the software controlling the car does not suffer from fatigue, intoxication, or distraction that human drivers are so prone to. Moreover, driving the car for hours might be a waste of time. This time can be used efficiently in a self-driving vehicles for all sorts of activities (office work, reading, online shopping, watching multimedia, etc.), and this is why the prospect of these use cases has

attracted many IT giants like Google and Apple to work in this field (Abraham et al. 2016). Both companies are interested in autonomous driving although the exact implementation of their business model might differ.

A race is on and lots of new cooperation models and alliances have come up (Dudenhöffer 2016; Freitag 2016).

In Table 11.2 the transition from driver assistance to autonomous driving is illustrated. One can differentiate five levels (1–5) of various degrees of assistance and a level 0 without any assistance (URL11 2015; URL21 2017). The first level (simple) includes assistance functions of active safety like ABS and ESP; the next level (partial automation) supports the driver with steering like lane keep assist, keeping the distance to the front car like adaptive cruise control or parking. Level 3 (conditional automation) already provides major assistance with maneuvering the car, long distance drive control, and remote parking features. However, the driver still has to supervise/monitor everything and from time-to-time resume control to show his or her attention. On level 4 (high automation), there are defined use cases like highway travel where the vehicle can drive automatically. Eventually, level 5—full autonomy—does not involve any human interference any more. The car does everything on its own, and the passenger can concentrate on other activities. Here the control of the car is completely handed over to the machine as shown in Table 11.2.

Level 5 autonomy means that the car will be able to take a passenger from one point to another without any manual intervention. It is clear that the step from level 3 (conditional automation) to 4 (high automation) and finally to level 5 (full automation) is not just a linear step but an abrupt jump of complexity with major impact on the whole automotive technical and legal ecosystem. Connectivity will play a major role as the self-driving car can process a huge amount of information and can get crucial warnings well in advance (e.g., about any obstacles on the road, even if it is behind a curve where it is not visible). Traffic signs will actively communicate with cars approaching, and connected cars can negotiate the way of right at crossings automatically. Cybersecurity becomes also crucial as autonomous cars will communicate with infrastructure, the cloud, and other cars (see also Chap. 6).

In order to make autonomous driving a reality, one needs a combination of different methodologies:

- Adaptive software systems that can be updated over the air and communicate with software systems in the clouds.
- Artificial intelligence and machine learning to actively learn and improve the performance.
- Car-to-infrastructure and car-to-car communication providing valuable information about the traffic situation.
- Different sensors (vision), with different capabilities, accuracy, and response times will generate a detailed picture of the environment.
- High-definition maps will give precise information about the surroundings, including detailed information in the third dimension (e.g., the elevation of a street border).

Table 11.2 Transition steps to autonomous driving (URL.11 2015)

Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
Driver only	Assisted	Partial automation	Conditional automation	High automation	Full automation
Driver continuously performs the longitudinal and lateral dynamic driving task	Driver continuously performs the longitudinal or lateral dynamic driving task	Driver must monitor the system at all times	Driver does not need to monitor the system at all times	Driver is not required during defined use case ^a	No driver required during entire journey
No intervening vehicle system active	Other driving task is performed by the vehicle	System performs longitudinal and lateral driving task in a defined use case ^a	Driver must be capable of resuming dynamic driving task	System performs the longitudinal and lateral dynamic driving task in all situations in a defined use case ^a	System performs entire dynamic driving task on all road types, speed ranges, and environmental conditions

^aUse cases refer to road types, speed ranges, and environmental conditions

- Powerful new bus systems to transport the increased multimedia sensor information.
- Scalable software architectures and middleware for processing sensor input, implementing fusion algorithms, pre- and post-processing, and analysis and machine learning.
- Semiconductor solutions for fast image and digital signal processing (Nvidia, Qualcomm, etc.).
- Sensor fusion algorithms to combine different sensor sources.

The fundamental aspects of advanced driver assistance functions that form the basis for autonomous driving are illustrated in Fig. 11.18 (Müller and Haas 2014; Reif 2014).

Camera sensors, image processing, and analysis will play a major role as these technologies have made huge progress over the last two decades and also are relatively cheap. Therefore, Sect. 11.3.5 gave insight into algorithms and their rapid prototyping in MATLAB/Simulink. Moreover, advances in mobile and autonomous robots have yielded a lot of results that are now valuable for self-driving cars (Bekey 2005; Corke 2011; Hertzberg et al. 2012; Kaplan 2016).

The OEMs and technology suppliers follow different strategies in implementing ADAS, high automation and full autonomy. In the case of adaptive cruise control, for example, some OEMs rely on a stereoscopic camera, while others use long-range radar in conjunction with a mono-vision camera (URL12 2015; URL13 2016; URL25 2017). Some highly automated vehicles may deploy three or more lidars in conjunction with additional sensors and GPS to give the vehicle a 360° view of its surroundings. Others might not use lidar at all, operating with a combination of radar and camera systems instead (URL5 2017). Whatever the specific choices might be, OEMs will rely on improved processing speeds to handle the large amount of data from the sensors.

The interior design of autonomous cars will differ substantially from classic cars as the driver's seat, steering wheel, and pedals are not needed any more. Instead the space can be used otherwise. The interior might look more like a living room where passengers can sit face-to-face in a meeting room atmosphere.

Automotive OEMs are experimenting with these additional degrees of freedom, and many exciting ideas have come up. In this regard, Fig. 11.22 shows VW's vision of a flexible interior design which was presented at the IAA 2017, and Fig. 11.24 shows Daimler's EQ vision. VW's concept car is both an autonomous vehicle as well as a "normal" car which is being driven by a human driver as it can be seen from Fig. 11.19 and Fig. 11.20. The steering wheel is retractable and can be pulled back if the car switches from human-driven by a human self-driving mode as shown in Figs. 11.21 and 11.22.

Such flexibility is very useful as the driver can opt for autonomous mobility after drinking alcohol, being tired or generally not feeling fit for driving. A car could also enforce the autonomous mode if it senses that the driver is intoxicated with alcohol.

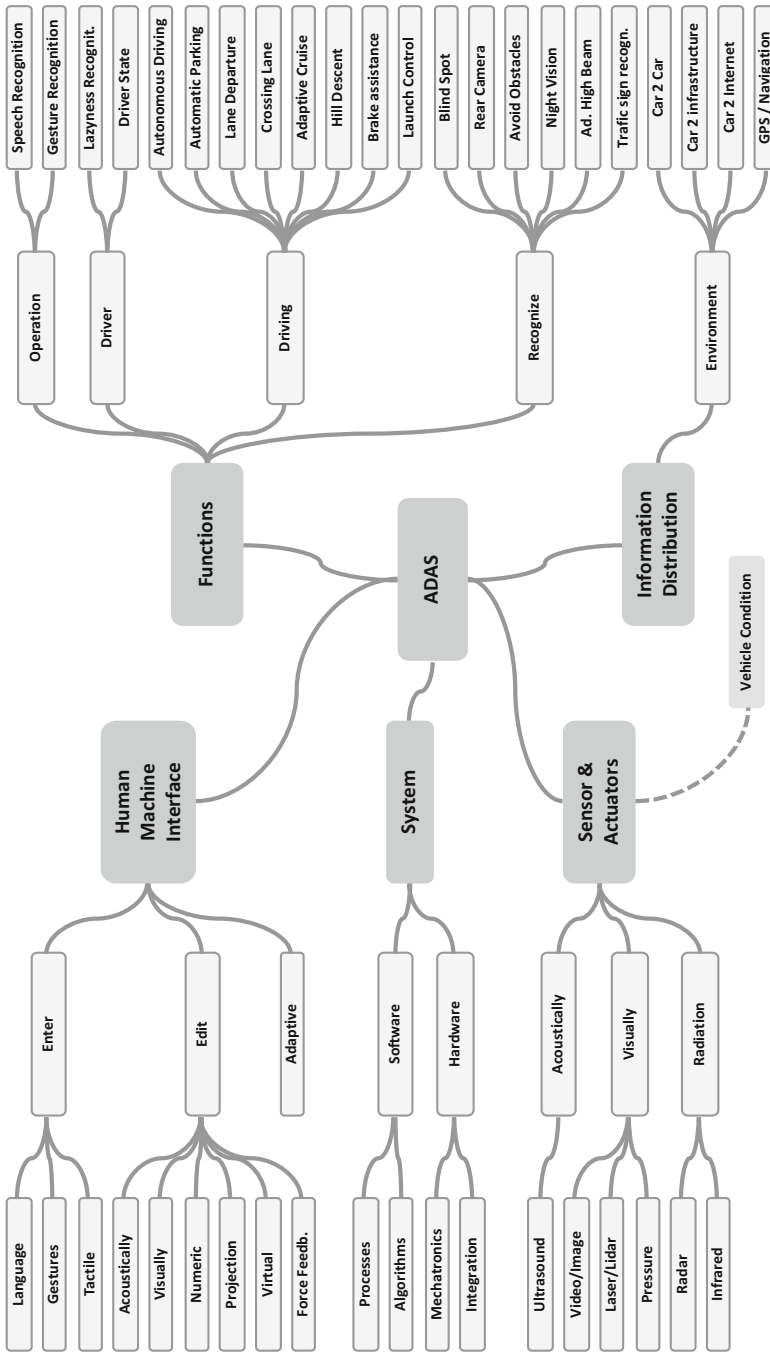


Fig. 11.18 Mindmap for ADAS and autonomous driving (Müller and Haas 2014)



Fig. 11.19 VW's study of an autonomous car at the IAA (URL11 2017)

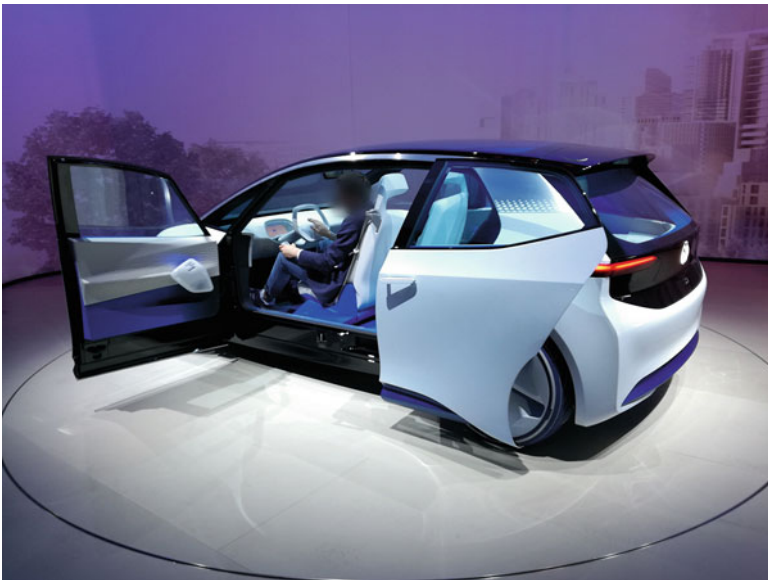


Fig. 11.20 Human driver uses a steering wheel to control lateral movements

Fig. 11.21 Steering wheel automatically retracts if one switches to autonomous driving mode



Fig. 11.22 If the car drives autonomously, the steering wheel is retracted and does not occupy space in the interior design

Fig. 11.23 Sensor on top of a car to map surroundings for HD maps



The key challenge for any automated driving system is to manage and combine the significant amounts of data coming from the different sensors and to create a consistent model from this data which can be used to make decisions about driving behavior. A common solution to this problem is the creation of hierarchical sensor fusion architectures as described in Balani (2015).

Most sensors are equipped with a dedicated processing unit that creates a digital representation of the raw, often analog sensor data. Figure 11.23 shows LIDAR sensors on top of a car which are used to generate high-definition 3D maps for TomTom (URL8 2017).

Sensor fusion combines the outputs of multiple sensors. For example, the data from two cameras can be combined to extract depth information (also known as stereo vision). Similarly, data from different sensor types with overlapping fields of view can be merged to improve object detection and classification and to create a more precise model (Balani 2015; URL5 2017).

It is also possible to add data from external systems which feed their data into the cloud giving access to detailed map, traffic, and weather data. The addition of data from a V2X gateway is also possible. The result is a detailed 3D map of the car's surrounding environment (Balani 2015). This map is object-based and includes lane markers, other vehicles, pedestrians, cyclists, street signs, traffic lights, and so on. The detailed map is embedded within a larger, less detailed map which is required for navigation. Both model perspectives are updated in real-time, although at different intervals creating a virtual reconstruction of the real world based on sensor data (Balani 2015).

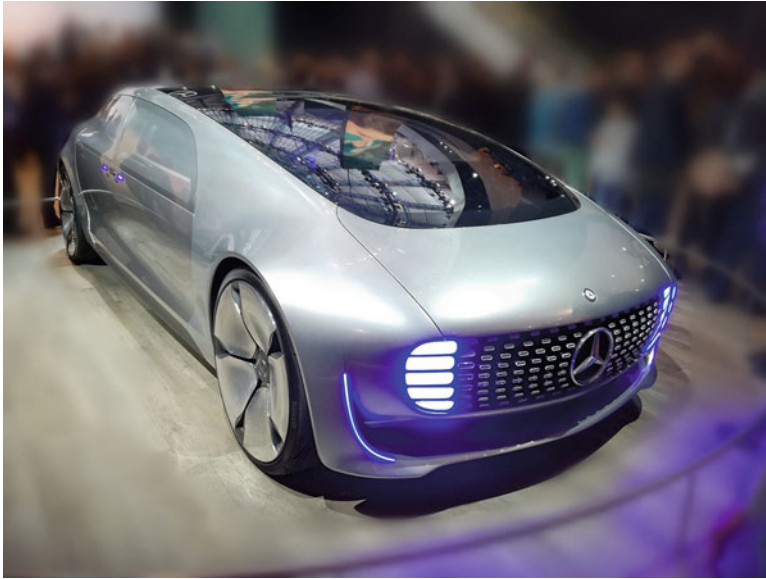


Fig. 11.24 Daimler's EQ vision of an electric and autonomous car

Short-range communications technology such as vehicle-to-vehicle and vehicle-to-infrastructure communication, collectively referred to as V2X, can be effectively applied to complex driving environments to enhance the safety of autonomous vehicles. V2X technology can supplement onboard sensors to gather and transmit environmental data, enabling the car to, for example, peer around corners and negotiate road intersections better than a human driver would (URL5 2017).

V2X technologies, which are today being developed in parallel with self-driving technologies, will enhance the performance and overall safety of autonomous vehicles (URL5 2017). The complexity of the driving environment will likely govern the introduction sequence of partially autonomous features as well. For instance, highways are less complex driving environment than urban streets or parking lots, which are full of non-standard infrastructure and involve a high level of interaction with other vehicles, pedestrians, and objects (URL5 2017).

Also, low-speed environments, such as traffic jams, may present fewer risks than high-speed driving scenarios. Clearly, autonomous driving requires huge investments and only a few companies can develop the technology in-house (Schaal 2017) which is one of the reasons why so many new alliances are coming up (Freitag 2016).

All these factors will influence the pace of adoption over the coming years. The technology will not gain commercial scale overnight; and, in fact, it may take several years before OEMs will be able to offer autonomous features at a price that is both acceptable to consumers and profitable for the manufacturer (Grünweg 2016; Maurer et al. 2015; URL11 2015). Form (2015) gives an overview of timelines and makes a prediction when the first fully autonomous cars can be seen on the road. The challenges can be clearly seen from the experience with Tesla's autopilot function (URL15 2016; Becker 2016).

11.5 Regulations, Public Acceptance, and Liability Issues

The previous section focused on technical challenges of autonomous vehicles. This section briefly deals with equally important aspects, like the regulatory frameworks, public acceptance, and liability issues (URL11 2015).

As with any new method of transportation, the regulatory environment plays a crucial role in its adoption. Public opinion has a significant impact too. In addition, there must be greater clarity on issues of liability. The incidents with Tesla have shown how sensitive the topic can be (Becker 2016).

11.5.1 Regulations and On-Road Approval

Autonomous driving on public roads is currently restricted by law. According to the Vienna Convention of Road Traffic (URL17 2017), which was ratified by over 70 countries as the foundation for international road traffic regulation, a driver must be present and in control of a moving vehicle at all times (URL8 2015). This clearly would not allow any self-driving vehicles to be deployed.

In May 2014, an expert committee of the United Nations added a new rule to the Vienna Convention: *Systems that autonomously steer a car are permissible if they can be stopped by the driver at any time.*

This recent addition represents a significant step forward in the development of automated driving, and several countries are now reviewing national legislation to allow self-driving vehicles in specific circumstances where automated technology has been proven to be sufficiently mature and safe. Early policy reviewers and therefore the countries that are likely to feature early adoption include the USA, UK, and New Zealand (URL8 2015).

It is important to note that the USA has not signed the Vienna Convention. Recent statements by the National Highway Traffic Safety Administration (NHTSA), like the response to Google's inquiry about self-driving cars, indicate a favorable position regarding self-driving technologies (URL10 2016; URL18 2017).

However, NHTSA admits that there is still a lot of work to be done in order to come up with a comprehensive legal framework for autonomous driving.

11.5.2 Toward a Statutory Framework for Autonomous Driving

The ministry of transportation in Germany and its counterparts in Europe are currently working on the statutory framework that would allow piloted and highly automated driving (Form 2015).

The governments still have to pass a comprehensive legislation. The framework is complex as it has to deal with various new issues like ethical dilemmas, etc. (Maurer et al. 2015).

Several traffic rules and regulations will need modifications:

- Road Traffic Act
- The highway or autobahn code
- Driving license regulations
- Road Traffic Licensing Regulations

Associations like VDA (German Automotive industry) and ACEA (European Automobile Manufacturers Association) are discussing the issue too. Both organizations have published white papers outlining their position on the matter (URL9 2016; URL16 2015).

11.5.3 Acceptance of Autonomous Driving and Ethical Difficulties

Several studies have attempted to measure public acceptance of autonomous driving, with substantial research undertaken in Germany and the USA (Maurer et al. 2015; URL8 2015). Findings suggest that people have mixed feelings about self-driving vehicles. On one hand there are many advantages, like saving valuable time, less accidents, and mobility for the elderly and visually impaired; and on the other hand, the technology has to mature, the legal framework is not yet clear, and self-driving cars will certainly have a disruptive effect on the logistics, transport, and mobility industry. Already, driverless trucks are being tested on the roads of Nevada, and ride-hailing companies experiment with autonomous taxis. If there are no adequate replacement jobs for drivers of trucks, taxis, busses, and so forth, there will be a strong opposition (Eckert 2016; Haas 2014).

Accidents due to system's limitations like Tesla's autopilot failure have attracted a lot of attention, even though humans might have failed in a similar scenario and human behavior is one of the main causes for severe accidents (Form 2015; Seeck 2015).

The biggest hurdle to public acceptance is probably ethics. For self-driving vehicles one has to define in detail how the vehicle will react in various situations, recognizing that passengers, other road users, and pedestrians could be hurt because of the self-driving vehicle's decision (URL8 2015).

As vehicles gradually become more automated, liability is a further concern that must be addressed. If a self-driving vehicle is involved in a road traffic accident, who is liable for the damage caused: the driver of the vehicle, the vehicle owner, or the manufacturer?

At present, liability is based on the premise that the person using the vehicle is responsible for its safe operation (URL8 2015).

According to current regulatory frameworks (see Maurer et al. 2015):

- Liability for damage to property and person is with the driver or vehicle owner.
- Liability for the vehicle—accountability for manufacturing errors including constructional defects, manufacturing defects, and faulty instructions -is with the manufacturer.

The key is a modification of traffic regulation stating that the driver does not violate his obligations and it can not be regarded as negligence if the control task is transferred to the system. This implies that the driver cannot be prosecuted according to criminal law if the accident is caused by technical failure. The financial liability will be taken care by the automobile liability insurance.

In this regard, it will be important to have devices that record any relevant information as there is no human driver to interrogate and report as a witness. This of course also raises data protection and privacy issues (Jung and Kalmar 2015; Reuter 2015).

Currently the insurance products for self-driving vehicles are, however, nonexistent (URL8 2015). In the context of a fully self-driving vehicle, liability has to be reconsidered. There can be no such thing as driver liability. And so long as the vehicle owner can prove correct maintenance of the vehicle, liability for damage to property and person shifts to the manufacturer (URL8 2015).

There are a couple of options for manufacturers to reduce their liability. The key question, however, is if a wrong decision on the part of the automatic controller could be considered to be a product defect, which would result in a liability of the OEM. Similar discussions are ongoing regarding robots—could one press charges against a robot?

The manufacturer may be able to insure its liability, and this will require the support of insurance providers convinced by the evidence that autonomous driving results in fewer, and less lethal, accidents than conventional driving (URL8 2015).

Clearly, it will be necessary to overcome obstacles of regulation, public acceptance, and liability can be deployed. Insurance companies are now recognizing the need to be prepared. By modifying and extending the existing terms of insurance, vehicle insurance companies may play a crucial role in accelerating the adoption of autonomous vehicles (URL8 2015).

11.5.4 Test on the Autobahn

Some countries have already addressed issues of regulation are allowing self-driving vehicles on their streets, at least for initial trials. The technology is seen a competitive advantage and nobody wants to stay behind. In the US, California and Nevada have passed special legislations for autonomous vehicles.

Also, Germany wants to be among the leaders in autonomous driving. The Ministry of Transportation and Digital Infrastructure (BMVI) has developed a comprehensive strategy that deals with all relevant questions and issues.

The digital test field “Autobahn” on the A9 in Bavaria is a test environment for experimenting with connected and automated driving, (see Fig. 11.25). The test track is open for all innovative companies, OEMs, suppliers, IT product companies, and research institutions. The goal is a certificate “Tested on German Autobahn” (URL15 2017). The test track is close to Audi’s headquarter in Ingolstadt. Figure 11.27 shows Audi’s concept car which was presented at the IAA 2017.

Test drives and the gathering of huge amounts of data are extremely important to validate higher levels of autonomous driving (Pickhard 2016).

Fig. 11.25 Germany allows testing of autonomous vehicles on a stretch of the A9 motorway



11.6 E/E Architectures and Middleware for Autonomous Driving

Higher-level ADAS functions and autonomous driving require a fundamentally different E/E architecture (Forster 2014; Hudelmaier and Schmidt 2013; Kern 2012; Lang 2015; Weiß et al. 2016). The size of program code required will increase significantly, and the amount of data transferred on the bus systems grows exponentially as vehicle manufacturers move from ADAS to partial autonomy, and finally to full autonomy (URL8 2016).

In Fig. 11.26 the E/E architecture and ECU topology of a modern car with various bus systems and automotive Ethernet are shown. The ADAS features form a special domain that is connected with a high-speed Ethernet bus system. Ethernet was developed in the computer network domain in the 1970s and has now become feasible for vehicle applications (e.g., camera-based ADAS) through the Broadcom R-Reach technology with unshielded twisted pair wires (Arndt 2015; Matheus and Königseder 2015).

Many proven technologies from computer networks, like TCP/IP, could be transferred from this field to automotive E/E and deployed in this context (Ernst 2016; Matheus and Königseder 2015; Schaal 2012; Schaal and Schwedt 2013; Weber 2013; Weber 2015).

ADAS sensors like cameras provide huge amounts of data, and the hard real-time requirements of higher levels of autonomy demand high throughput, short latencies, and a high degree of flexibility.

On the CAN bus the signals are passed to all connected ECUs, regardless if the information is relevant or not (Streichert and Traub 2012). Modern automotive E/E architectures rely on high-speed, scalable bus systems and a so-called middleware which handles the complex interaction between the different communication partners (sensor nodes, electronic control units and actuators).

ADAS systems requires a carefully designed, reusable, and scalable automotive software system which can be achieved by service-oriented architectures (SOA). The transition to autonomous driving will generate an even bigger need for software architecture. Currently, there is a lot of research going on in the field of ADAS software architectures (Fürst 2016; Lamparth and Bähren 2014; Thiele et al. 2013; Wagner 2015 and URL12 2016).

Service orientation is a well-accepted standard in classical software development (Balzert 2011; Schäfer 2010). It offers interesting possibilities to partition and structure ADAS software functions (Wagner 2015). To enable bandwidth efficiency, automotive IP networks—in contrast to static CAN communication—are set up in a dynamic and service-oriented way (Schaal 2012).

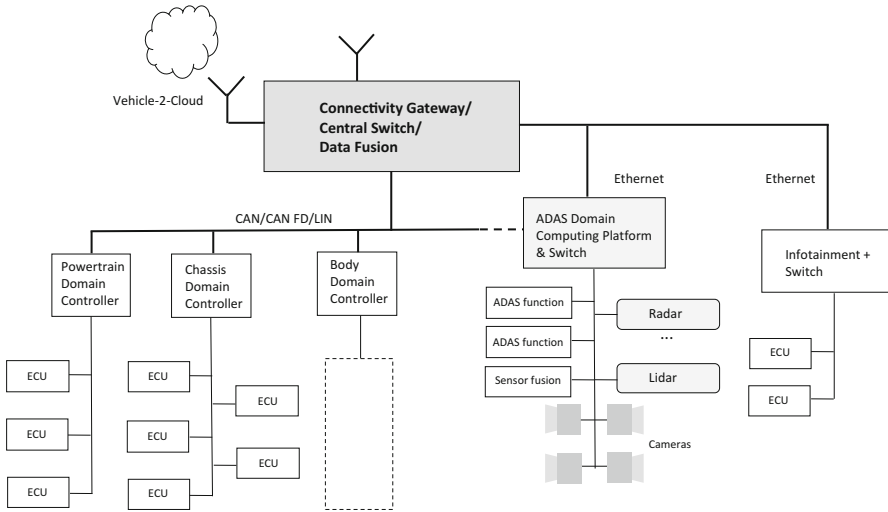


Fig. 11.26 A vehicle E/E architecture for ADAS and autonomous driving



Fig. 11.27 Audi presented a concept car for autonomous driving at the IAA 2017

The middleware plays a very important role in the implementation of sophisticated ADAS functions and autonomous driving capabilities. Figure 11.28 illustrates a typical ADAS software architecture. The top layer consists of various ADAS functions like LKA and AEB. These software modules communicate with a

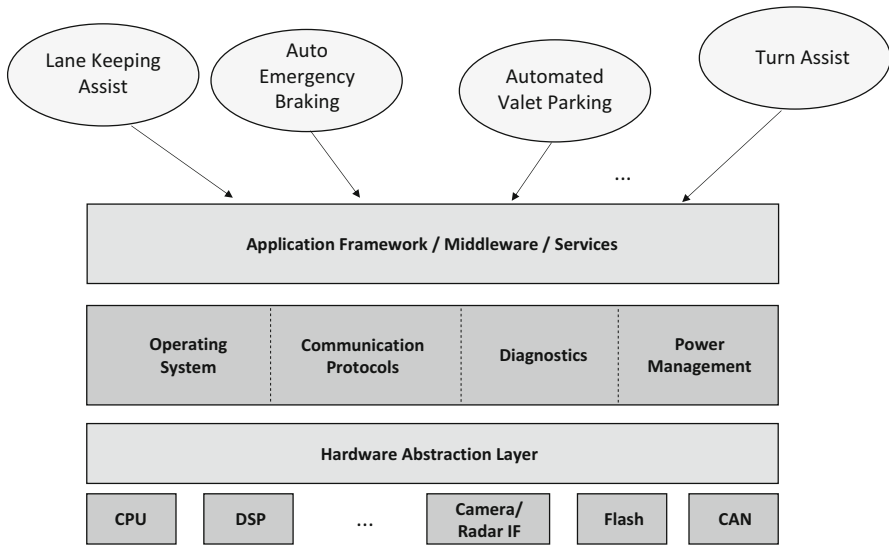


Fig. 11.28 Middleware for ADAS functions

middleware that orchestrates the interaction and takes care of managing the information flow. The next layer consists of the operating system drivers, communication protocols, diagnostics, power management and core algorithms for signal processing and sensor fusion. The actual hardware forms the lowest layer. This layer is encapsulated by a hardware abstraction layer (HAL); therefore that the functional layers do not interact with the hardware directly (Tanenbaum and Bos 2015).

There are many middleware concepts which are studied in the computer science field of distributed systems (Schill and Springer 2012; Silberschatz et al. 2010; Tanenbaum and Van Steen 2017). However, the utilization in the automotive domain is restricted by cost, performance, reliability, functional safety, and real-time constraints. The middleware for ADAS and autonomous driving functions should carefully balance the traffic on the network, using broadcast, public, and subscribe filters to restrict the distribution of unnecessary information.

BMW has developed and specified Scalable Service Oriented Middleware over IP (SOME/IP), which is an open protocol that fulfills automotive requirements offering publish and subscribe and remote procedure call mechanisms (Matheus and Königseder 2015; URL11 2016). Figure 11.29 illustrates the concept. The SOME/IP middleware uses a TCP/IP stack on an Ethernet connection (BroadR-Reach or T-Base (Arndt 2015)). If a software component or application needs to call a remote function across ECU boundaries, the middleware will establish a TCP/IP-based client/server communication.

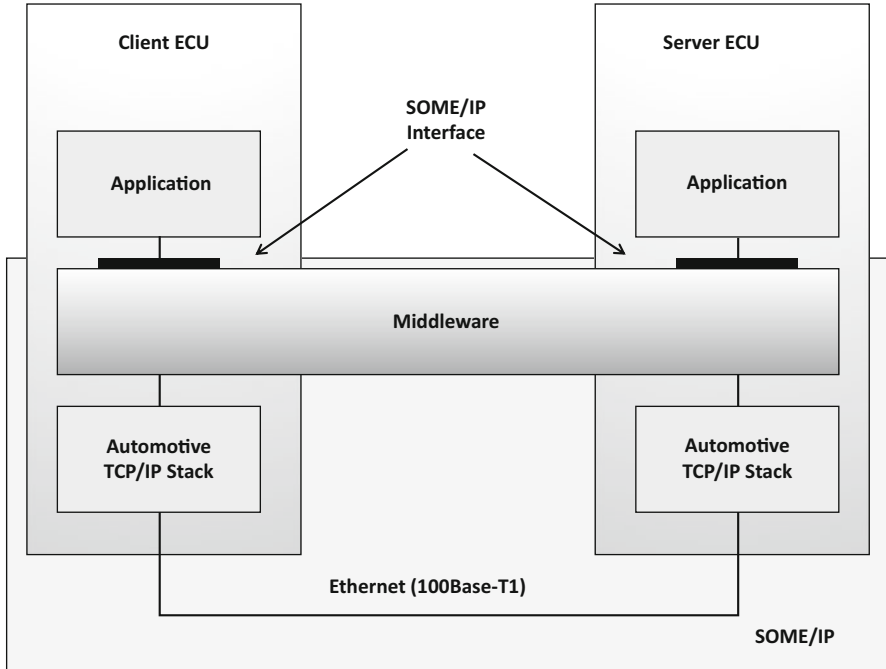


Fig. 11.29 SOME/IP (URL11 2016)

SOME/IP offers interfaces for service-oriented communication. This distinguishes it from the pure signal-based broadcast communication systems like CAN (URL9 2015; URL11 2016).

SOME/IP interaction is roughly subdivided into three areas: service discovery (SD), remote procedure call (RPC), and access to process data. SD lets ECUs find services or offer their services in the network which are accessed via RPC as shown in Fig. 11.30. In addition to that, it is possible to set up notifications for specific events.

Figure 11.30 depicts the communication pattern in SOME/IP. The RPC mechanism allows a classic remote procedure call across the communication network. It abstracts from all details like finding the server process/application, and managing the data transfer. Apart from this, there is a publish/subscribe mechanism which notifies applications if an event is detected. Only those applications that have been registered will be informed. This allows for a careful utilization of bandwidth. In the context of ADAS or autonomous driving, it could mean, for example, a specific function would only register for those events which are relevant and would only be notified if the corresponding sensors would have generated such an event. Publish/subscribe mechanisms are also very helpful if the system is being reconfigured, or in the scenario of a graceful downgrading of subsystems which do not work properly and have to be shut off (Matheus and Königseder 2015).

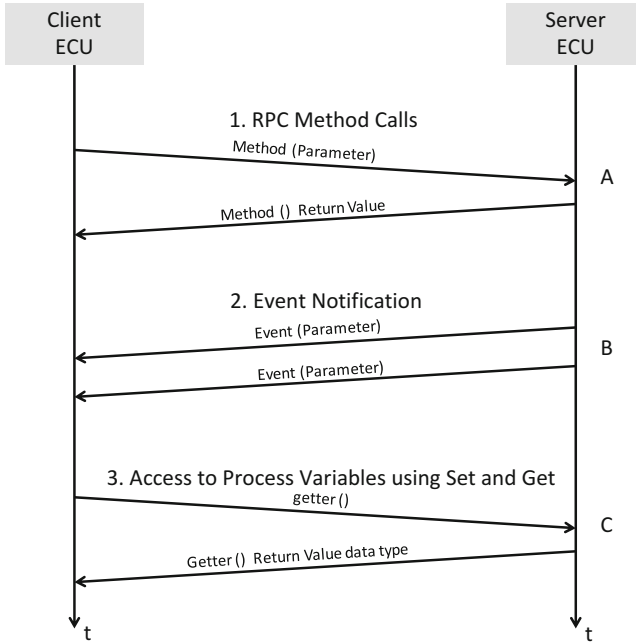


Fig. 11.30 Communication in SOME/IP (URL11 2016)

Besides SOME/IP, which is already an integral part of AUTOSAR, several other middleware standards are currently being used in autonomous cars, chief among them (URL10 2015; URL11 2016; URL19 2017; URL20 2017):

- Data Distribution Service (DDS)
- Automotive Data and Time-Triggered Framework (ADTF)
- Robot Operating System (ROS)
- Common API (GENIVI/Open Source)

Data distribution service (DDS) is a middleware standard which was defined by the Object Management Group (OMG). The company or vendor RTI provides an implementation which they call Connex DDS, specifically for applications in autonomous driving (URL20 2017).

The ADTF framework is very popular in Germany, for example in Volkswagen and Audi. ROS was developed as a standard operating system for robots. It supports many robot-specific features, like standard message definitions for robots, robot geometry library, robot description language, preemptable remote procedure calls, diagnostics, localization, mapping, and navigation (URL19 2017; URL20 2017) which can be directly used in autonomous driving.

The AUTOSAR standardization committee is currently reexamining communication middleware (Fürst 2016; Weber 2013; URL23 2017) to embed further middleware standards in the AUTOSAR stack.

11.7 Cybersecurity and Functional Safety

ADAS and autonomous driving are potential targets for cyberattackers, with potentially far reaching consequences.

Cybercriminals and hackers could exploit various cyberattack surfaces like:

- In-vehicle Infotainment System (IVI)
- Telematic Control Unit (TCU)
- V2X communication infrastructure (V2V, V2I)
- Connected smartphones and mobile apps
- Connection between smartphone/key and vehicle
- Software stacks, e.g. middleware (exploiting design flaws, backdoors, etc.)
- Over-the-air updates of software and firmware (SOTA, FOTA)
- ADAS sensors
- HW/SW supply chain (backdoors, spy chips, HW vulnerabilities, software flaws, etc.)
- Data recording devices
- Backend systems
- GPS devices (compromising the localization)
- OBD-II port/remote diagnostics
- Electric powertrain (battery systems, charging infrastructure, communication protocols between charging infrastructure and vehicle)

The complexity of the software in an autonomous vehicle creates a special threat which is well known from operating systems. There might be design flaws that are known to a few developers, so-called zero day vulnerabilities, which could be exploited for a zero-day attack.

Also the middleware could be a target for cyberattacks. In this regard, Herold et al. (2016) and Wolf et al. (2015) discuss cybersecurity issues of SOME/IP.

Another mode of attack, which is especially harmful for self-driving cars, is the direct attack on sensors like blinding cameras, confusing camera auto control, relaying or spoofing signals.

Malware could be introduced through various ports of the network and could compromise the attached subsystems, actuators, and sensors. In such a scenario, vehicles could receive false signals, intentionally misguiding them.

Autonomous cars rely on sophisticated machine learning technologies. If one knows the algorithms, their sensitivity to noise and potential vulnerabilities, this could be exploited, increasing the overall risk.

Cybersecurity flaws will have an immediate impact on the functional safety.

Functional safety is described by specific automotive safety integrity levels (ASIL) (URL4 2016). This was discussed in detail in Chap. 4. The ASIL scheme distinguishes four ASIL levels, A to D, according to the severity of a failure (URL5 2016). The functioning of the rear camera, for example, refers to ASIL B, anything affecting the breaks is ASIL D.

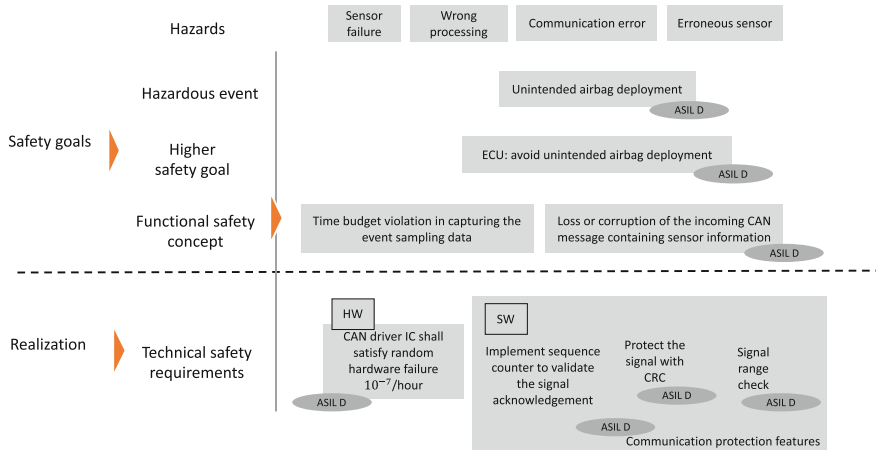


Fig. 11.31 Functional safety and automotive software (see Vivekanandan et al. 2013)

KPIT has developed a framework for handling safety goals (Vivekanandan et al. 2013). Figure 11.31 illustrates this approach for the event of unintended airbag deployment if a special child seat is present. The safety goal (SG) is to prevent such an unintended inflation of the airbag which is classified as a severe event falling into the category of ASIL level D. The functional safety concept is to prevent and mitigate a loss of the incoming CAN message that contains this information. On the implementation level, there are HW and SW requirements to fulfill. The CAN driver has to operate within a specific failure mode for random hardware failure and the signals have to be checked for range and integrity, for example, by a cyclic redundancy check (CRC). Furthermore, a sequence counter will validate that the signal is acknowledged in a given time frame.

A cyberattack on the CAN bus could scramble up and delay such a signal too and lead to wrong values. Hence, a similar approach could be employed to check the validity and timing of the signal value, even though there is no hardware failure or glitch in timing but a cyberattack that leads to functional safety concerns.

In Fig. 11.32 the concept of simultaneous engineering of cybersecurity and functional safety is shown which has been proposed by various researchers (Nause and Höwing 2016; Serio and Wollschläger 2015; URL26 2017) as cybersecurity breaches can have an immediate effect on the functional safety of automotive systems (Klauda et al. 2015; Solon 2015).

This is why it is crucial to integrate cybersecurity as a design goal into the automotive R&D process (Haas and Möller 2017; Mahaffey 2015a, b; Nause and Höwing 2016; Sushravya 2016). This approach is also reflected in the comprehensive framework of Fig. 11.33.

As outlined in the previous sections, autonomous driving requires a close integration of information from various sources, for example, sensors, infrastructure, other cars, cloud, and so forth. The reliability, availability, and cyber-safety are

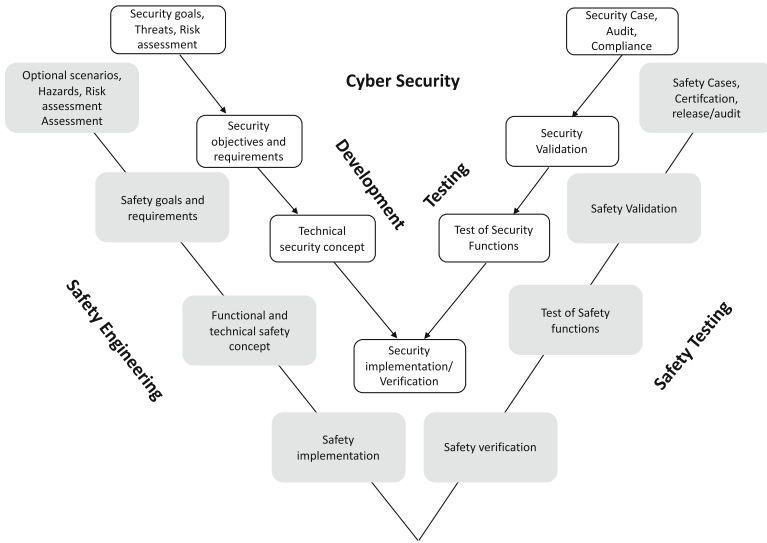
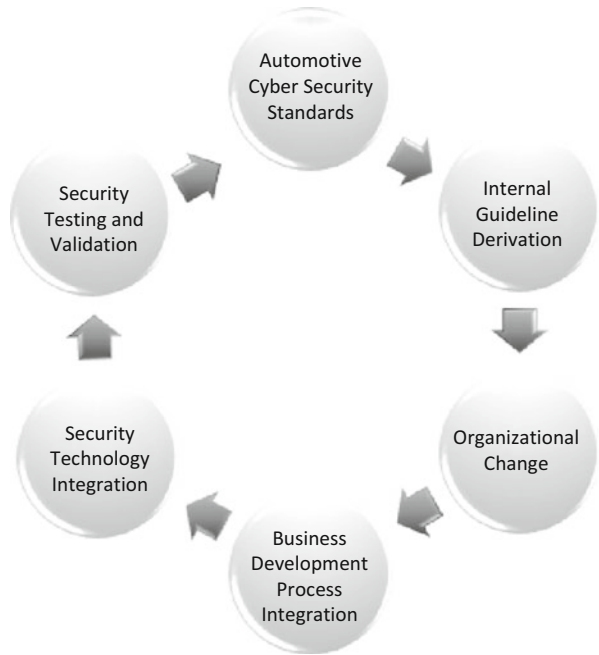


Fig. 11.32 Functional safety and cybersecurity (see Klauda et al. 2015; URL26 2017)

Fig. 11.33 Cybersecurity needs a holistic approach



important design aspects. Serio and Wollschläger (2015), Wolfsthal and Serio (2015) and Currie (2015) analyze the cyber threats to connected cars and suggest various solutions for intrusion detection.

Cybersecurity for autonomous cars can only be achieved with a holistic and integrated approach (Pickhard et al. 2015; Serio and Wollschläger 2015; Weimerskirch 2016) as discussed in Chap. 6 where one has to look at hardware (HW) failures, software (SW) vulnerabilities with regard to cybersecurity and system safety simultaneously. Reuter (2015) argues that functional safety is closely related with the security of data in a modern car and warns of the liability consequences if this topic is not dealt with properly.

If a cyber attack happens involving a new malware or attack vector, it is important to react swiftly and to update the signatures of intrusion detection and prevention systems as soon as possible. In this regard, over-the-air updates and security operation centers are crucial to fight cyberattacks (Zetter 2015; Brisbourne 2014). Unfortunately, over-the-air updates themselves are prone to attacks and require sophisticated cryptographic methods and key management to avoid security breaches.

11.8 Summary, Conclusion, and Recommended Readings

This chapter discussed *advanced driver assistance* functions and *autonomous driving*. ADAS has been around for a while. Adaptive cruise control, emergency breaking, blind spot detection, lane departure warning and lane keeping assist, as well as remote and automatic parking are just a few of many ADAS functions which make a car safer and easier to drive. As an example, we discussed the lane keep assistance function in a Seat Leon car.

Camera sensors are an inexpensive yet efficient way of sensing the environment. The chapter gave an overview of image processing and analysis for camera-based advanced driver assistance systems and showed how these algorithms can be rapidly implemented with MATLAB/Simulink. Automatic code generation is also possible on various target platforms. Hardware-in-the-loop (HIL) systems have evolved to test ADAS functions, and they will play an important role in the reliability analysis of higher automation and autonomous driving.

Non-functional requirements in ADAS like flexibility, modularization, responsiveness, reliability, testability, security, and so forth have had an impact on automotive E/E and software architectures. Middleware technologies like SOME/IP provide the flexibility and scalability needed. SOA and Message-oriented middleware (MOM) help to carefully utilize the bandwidth of the bus system.

The legal framework for autonomous driving has to still evolve, but there is a lot of pressure as countries do not want to fall behind in such an innovative domain. The key is the modification of the Vienna Convention which requires a human to be in charge at any time. Also, ethical issues have to be dealt with, and liability and insurance models have to adapt.

Autonomous driving is based on connectivity, complex software systems, over-the-air updates, and vehicle-to-backend communication. All this implies complex attack surfaces.

Functional safety and cybersecurity are intimately connected. Reliable autonomous cars will only be possible by adopting a “design by security approach” as has been proposed by various researchers (Haas and Möller 2017; Pickhard et al. 2015; Weimerskirch 2016; URL1 2015). Furthermore, cybersecurity for ADAS and autonomous driving requires a holistic approach.

11.8.1 Recommended Reading

Form (2015) outlines different developments in autonomous driving.

Markoff (2016) and Menn (2016) discuss the current interest in artificial intelligence in the context of drones, robots and autonomous cars, a research field that attracts huge fundings.

More information on software architectures for drivetrain control can be found in Orth et al. (2014).

The books (Silberschatz et al. 2010) and (Tanenbaum and Bos 2015) give a good overview of middleware from an operating system perspective.

Vahid and Givargis (2001) discuss the HW implementation of image processing in a case study of a digital camera.

Steinmüller (2008) is a compact introduction to the mathematics of image processing and image analysis.

Köncke and Buehler (2015) discuss the impact of cyberattacks on the industry in general.

The huge investments into autonomous driving can only be handled by a view OEMs (Schaal 2017), most of them need help from partners and strike new alliances. Eckl-Dorna (2016) looks at the cooperation between FCA and Google. More information on the Apple Titan project can be found in Sorge (2017).

Hunt et al. (1996) present a control algorithm for the lateral control of autonomous vehicles based on a network of local model (operating regime)-based controllers. A detailed overview of modeling, simulation, and control of vehicle driveline is given in Kiencke and Nielsen (2005).

For more information on digital control and adaptive control refer to Astrom and Wittenmark (1996), Ogata (2004), and Dorf and Bishop (2010).

Prostinett and Schimansky show how vulnerable we have become to the availability of the cloud discussing the partial shutdown of the AWS infrastructure in March 2017. This is also important for ADAS functions and autonomous driving, as some algorithms might run in the cloud.

Reuss et al. (2015) discuss the synergies between electro-mobility and autonomous driving.

Paar (2015) emphasizes the importance of cryptography to protect automotive E/E architectures. URL5 (2017) refers to the Spy Act which has triggered a lot of activities in automotive cybersecurity in the US. URL1 (2015) and URL 4 (2015),

(URL7 2015), and (URL16 2016) provide a good introduction to the field of automotive cybersecurity.

11.9 Exercises

- What ADAS functions are you aware of, and how would you categorize them?
- What are the different levels of driver assistance?
- What is the difference between conditional automated and highly automated driving?
- What research initiatives in autonomous driving are you aware of?
- Who are the leading suppliers in ADAS systems from your viewpoint?
- Who will cooperate with whom (OEM, suppliers, service providers, etc.)?
- What are the biggest hurdles for autonomous driving from a technology viewpoint?
- What sensors are important in ADAS?
- What is the relationship between autonomous/piloted driving and automated valet parking? What are the commonalities and what are the differences?
- What role do maps play in autonomous driving? Who will provide these maps? What are the collaboration models?
- Which R&D processes and safety guidelines (e.g., ISO 26262, ASIL) are being used in ADAS development?
- What associations, bodies, and committees are dealing with the regulatory and standardization framework of ADAS and autonomous driving?
- What are the biggest obstacles for autonomous driving from a legal viewpoint?
- What legal framework is applicable for autonomous driving? What modifications are needed in the future?
- What are the current state laws and legislative activities? What federal or state liability legislation is needed?
- What is the Vienna Convention on Road Traffic?
- What are the recent modifications to this framework which will lead the way to highly automated driving?
- What liability questions arise in autonomous driving?
- What are the biggest safety concerns regarding ADAS and autonomous driving?
How will safety be handled?
- What safety standards are applicable? How does ADAS safety differ from safety for piloted driving?
- What are the social implications of autonomous driving?
- Will people still want to drive themselves if autonomous cars are widely available?
- In which countries will autonomous driving be introduced first?
- In what of the following areas will autonomous driving be deployed first (logistics, public transport, individual transport)?
- What role will autonomous driving play for carsharing, ridesharing, electric cars?
- Compare the different strategies for autonomous driving in high-tech companies like Google, mobility service providers like Uber, and classical automotive OEMs.
Read about the AUTCLES project and write a short report about it.

What is the relationship and what are the synergies between autonomous driving and electro-mobility?

Comment on Tesla's approach to ADAS and autonomous driving.

Comment on the importance of cybersecurity in autonomous driving?

What is the relationship between functional safety and cybersecurity?

What are the biggest challenges in autonomous driving?

What role does V2X communication play for autonomous driving?

What are typical mobility use cases for autonomous driving?

Comment on the costs of autonomous driving. Will it be affordable?

Please compare OEMs strategies and announcements in autonomous driving.

How do Indian automakers position themselves?

What are the synergies between robotics and autonomous driving? What can we learn from other industries, e.g., aircrafts?

Please write a short report on potential cyberattacks on autonomous cars.

What are the most critical cybersecurity vulnerabilities of autonomous cars?

What solutions are available to make autonomous cars cybersecure?

Review the recent reports about cyberattacks on cars. What is the relevance for cybersecurity of autonomous cars?

What impact on traffic accidents will autonomous cars have?

Please comment on the complexity of autonomous driving from a HW and SW perspective.

What are the major challenges for the engineers of autonomous cars?

What impact will autonomous cars have on jobs in the transport business?

How do insurances prepare for autonomous driving?

What happens if autonomous cars move from one country to another?

Describe the ethical dilemma situation with driverless cars.

Please comment on the reliability of sensors and computational elements, aging, and wear and tear in ADAS systems.

What does the term robotification mean?

What impact does autonomous driving have on the interior design of a car?

References and Further Readings

(Abraham et al. 2016) Abraham, B., Brugger, D., Strehlke, S., Runge, W.: Autonomous Driving – Only a Trojan horse of Digital Companies?, ATZ elektronik, 01/2016

(Alheeti et al. 2015a) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An intrusion detection system against malicious attacks on the communication network of driverless cars. In: 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Pages 916–921, 2015

(Alheeti et al. 2015b) Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D.: An Intrusion Detection System against Black Hole Attacks on the Communication Network of Self-Driving Cars. In: 6th International Conference on Emerging Security Technologies (EST), Pages 86–91, 2015

(Arndt 2015) Arndt, C.: Developments of Automotive Ethernet Technologies - Introduction to the BroadR-Reach Technology and beyond, Continental & VDI Wissensforum, 06/2015

- (Astroem and Wittenmark 1996) Astroem, K., Wittenmark, B.: Computer Controlled Systems. Prentice Hall, Information and Systems Series, 1996
- (Balani 2015) Balani, N.: Enterprise IoT – A Definite Handbook, self-published, Kindle Edition, 2015
- (Balzert 2011) Balzert, H.: Textbook of Software Engineering: Design, Implementation, Deployment and Operation (in German), Springer Spektrum Publ., 2011
- (Beck 2016) Beck, T.: Do we need Autonomous Driving? (in German) elektronik.net, 01/2016, pp. 48–49. 01/2016
- (Becker 2016) Becker, J. Autopilot of Tesla – in a Tesla the risk is a standard feature (in German). Süddeutsche Online. November 17th 2016. Available from: <http://www.sueddeutsche.de/auto/autopilot-von-tesla-bei-tesla-ist-das-risiko-serienmaessig-1.3252192>
- (Bekey 2005) Bekey, G. A.: Autonomous Robots, Massachusetts Institute of Technology, 2005
- (Besenbruch 2014) Besenbruch, D.: Electronic Systems – Protection of manipulation, ATZ elektronik, 2014
- (Beynon et al. 2003) Beynon, M., Hook, D., Seibert, M., Peacock, A., Dudgeon, D.: Detecting Abandoned Packages in a Multi-camera Video Surveillance System. IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2003
- (Bhattacharjee 2013) Bhattacharjee, S.: Efficient Algorithm for Crossing Alert in Camera-based Advanced Driver Assistance Systems, Master of Technology Thesis, International Institute of Information Technology Bengaluru (IIIT-B), 2013
- (Bose 2004) Bose, T.: Digital Signal and Image Processing, John Wiley and Sons, 2004
- (Bridges 2015) Bridges, R.: Driverless Car Revolution Buy Mobility – Not Metal. Self-published, 2015
- (Brisbourne 2014) Brisbourne, A.: Tesla's Over-the-Air Fix: Best Example Yet of the Internet of Things? Wired online. February 2014. Available from: <http://www.wired.com/insights/2014/02/teslas-air-fix-best-example-yet-internet-things/>
- (Chucholowski and Lienkamp 2014) Chucholowski, F., Lienkamp, M.: Teleoperated Driving – Secure and Robust Data Connections, ATZ elektronik, 01/2014
- (Corke 2011) Corke, P.: Robotics, Vision, and Control, Springer Publ., 2011
- (Currie 2015) Currie, R.: Developments in Car Hacking. <https://www.sans.org/reading-room/whitepapers/internet/developments-car-hacking-36607>, 2015
- (Davies 2012) Davies, E. R.: Computer and Machine Vision: Theory, Algorithms, and Practicalities, Elsevier Publ., 2012
- (Dorf and Bishop 2010) Dorf, R.C., Bishop, R.H.: Modern Control Systems, Pearson Education, 2010
- (Dudenhöffer 2016) Dudenhöffer, F.: Who will be put in the fast lane (in German). Campus Publ., 2016
- (Eckert 2016) Eckert, D.: Robots will destroy millions of jobs. Welt online. 27th August 2016. Available from: <http://www.welt.de/wirtschaft/article157872907/Roboter-werden-Millionen-Jobs-vernichten.html?config14print#.08.2016>
- (Eckl-Dorna 2016) Eckl-Dorna, W.: Savior instead of aggressor: Fiat Chrysler courts Google (in German). Manager Magazin online, April 29th 2016. Available from: <http://www.managermagazin.de/unternehmen/autoindustrie/roboter-auto-allianz-warum-fiat-chrysler-mit-google-kooperieren-will-a-1090052.html>
- (Elgammal et al. 2000) Elgammal A., Harwood D., Davis L.: Non-parametric Model for Background Subtraction. In: Proceedings of the 6th European Conf. on CompVision-Part II, Pages 751–767, 2000
- (Elgammal et al. 2003) Elgammal A., Duraiswami R., Davis, L.S.: Efficient Kernel Density Estimation Using the Fast Gauss Transform with Applications to Color Modeling and Tracking. In: IEEE Transactions on Pattern Analysis and Machine Intelligence; Vol. 25 No. 11, 1499–1504, 2003
- (Ernst 2016) Ernst, R.: Automotive Ethernet – Opportunities and Pitfalls, Institut für Datentechnik und Kommunikationsnetze, ETFA 09/2016, Berlin, 2016

- (Fallstrand and Lindstrom 2015) Fallstrand, D., Lindstrom, V.: Automotive IDPS: Applicability analysis of intrusion detection and prevention in automotive systems. Master' Thesis. Chalmers University of Technology. Available from: <http://publications.lib.chalmers.se/records/fulltext/219075/219075.pdf>
- (Form 2015) Form, T.: Autonomous Driving – Quo vadis?, ATZ elektronik, special edition, 07/2015
- (Forster 2014) Forster, F.: Development Embedded Systems, ATZ elektronik, Vol 9, 01/2014, Pages14–18, Springer Vieweg, Springer 2014
- (Freitag 2016) Freitag, M.: Robotic cars – German manufacturers in pole position (in German). Manager Magazin online. July 26th 2016. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/roboterautos-deutsche-autobauer-fuehrena-1104783.html>
- (Fürst 2016) Fürst, S.: AUTOSAR Adaptive Platform for Connected and Autonomous Vehicles, In: EUROFORUM Elektronik-Systeme im Automobil, 02/2016
- (Gaonkar et al. 2011) Gaonkar, P., Nanthini, S., Manoj, S., Mamilla, S.: Lane Departure Warning System, Class Paper, Car IT and Cybersecurity class, IIIT-B, 2011
- (Giachetti et al. 1994) Giachetti, A., Campani, M., Torre, V.: The use of optical flow for the autonomous navigation, In: Proc. 4th Euro. Conf. Comput. Vision, 1994
- (Grünweg 2016b) Grünweg, T.: Ford strategy – Autonomy for All (in German). Spiegel online. October 11th 2016. Available from: <http://www.spiegel.de/auto/aktuell/ford-plant-roboter-taxi-flotte-wie-uber-a-1114025.html>
- (Gonzalez et al. 2008) Gonzalez R. C., Woods, R. E., Eddins, S. L.: Digital Image Processing Using MATLAB, Pearson Education, New Delhi, India, 2008
- (Gonzalez and Woods 2008) Gonzalez, R. C., Woods, R. E.: Digital Image Processing, 3rd Edition. Pearson/Prentice Hall Publ., 2008
- (Haas 2014) Haas, R.: Socio-Economic Impact of Autonomous Driving in Emerging Countries, India as example, European Radar Conference, EuRad, Rome, 2014
- (Haas and Möller 2017) Haas, R. E., Möller, D. P. F.: Automotive Connectivity, Cyber Attack Scenarios and Automotive Cyber Security. Proceed. IEEE/EIT 2017, pp. 635-639, ISBN: 978-1-5090-4767-3/17
- (Hanselman and Littlefield 2008) Hanselman, D., Littlefield, B.: Mastering MATLAB 7, Pearson Education, India, 2008
- (Haykin 2009) Simon Haykin, Neural Network and Learning Machines, 3rd Ed., Pearson Education, Upper Saddle River, NJ, 2009
- (Herold et al. 2016) Herold, N., Posselt, S.-A., Hanka, O., Carle, G.: Anomaly Detection for SOME/IP using Complex Event Processing, Chair of Network Architectures and Services, Technical University Munich (TUM), Department of Computer Science, 2016
- (Hoffmann 2008) Hoffmann, D.: Software-Quality, Springer Publ., 2008
- (Horn and Schunk 1981) Horn K.P., Schunck, B.G.: Determining Optical Flow, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A., Pages 185-203, 1981
- (Hertzberg et al. 2012) Hertzberg, J., Lingemann K., Nüchter, A.: Mobile Robotics – An introduction from a computer science perspective (in German), Springer Vieweg Publisher, Berlin Heideberg, 2012
- (Hudelmaier and Schmidt 2013) Hudelmaier, P., Schmidt, K.: Chip Solutions For Driver Assistance Systems, ATZ elektronik 03/2013, Pages 48–52
- (Hunt et al. 1996) Hunt, K. J., Haas, R., Kalkkuhl, J.: Local Controller Network for autonomous vehicle steering, Control Engineering Practice, 1996
- (Jain 2000), Jain, A. K.: Fundamentals of Digital Image Processing, Prentice Hall Publ., 2000
- (Javed et al. 2002) Javed, O., Shah, M.: Tracking and object classification for automated surveillance, In: Proc. of ECCV, Pages 343–357, 2002
- (Johri 2016) Johri, S.: Attack Surfaces in Connected Cars, Class paper, Car IT and Cybersecurity class, IIIT-B, 2016
- (Johanning and Mildner 2015) Johanning, V., Mildner, R.: Car IT Compact, Springer Publ., 2015

- (Joshi 2009) Joshi, M. A.: Digital Image Processing - An Algorithmic Approach, PHI Learning, New Delhi, 2009
- (Jung and Kalmar 2015) Jung, C., Kalmar, R.: Re-interpret Data Security – the Data Gold and Business Models, ATZ elektronik, 04/2015
- (Kaplan 2016) Kaplan, J.: Artificial Intelligence, Oxford University Press, 2016
- (Karmann et al.1990) Karmann K.-P., Brandt A.: Moving object Recognition using and adaptive background memory. In: Time-Varying Image Processing and Moving Object Recognition, Pages 289–307. V. Cappellini, Ed: Elsevier Science Publishers, 1990
- (Kern 2012) Kern, A.: Ethernet and IP for Automotive E/E-Architectures – Technology Analysis, Migration Concepts and Infrastructure. Ph.D. Thesis, University of Erlangen-Nürnberg, Available from: <https://pdfs.semanticscholar.org/8106/455c487acc052cc701f48615f1172029a057.pdf>, Erlangen, 2012
- (Kiencke and Nielsen 2005) Kiencke, U., Nielsen, L.: Automotive Control Systems: For Engine, Driveline, and Vehicle. Springer Publ., 2005
- (Klauda et al. 2015) Klauda, M., Schaffert, M., Logospiris, A., Piel, G., Kappel, S., Ihle, M., Setting the Course for 2020 – change of paradigms in E/E architecture. ATZ elektronik, Pages 17–22, Springer Vieweg Publ., 02/2015
- (Köncke and Buehler 2015) Köncke, F. C., Buehler, B. O.: Cyber Attacks – Underestimated Risk for the German Industry (in German). Wirtschaftswoche online. November 9th 2015. Available from: <https://www.wiwo.de/technologie/digitale-welt/cyber-angriffe-unterschaetztes-risiko-fuer-die-deutsche-industrie/12539606.html>
- (Lang 2015) Lang, M.: High Degree of Integration of ADAS Functions into One Central Platform Controller, ATZ elektronik, Vol 10, 04/2015, Pages 40–43, Springer Vieweg Publ., 2015
- (Lamparth and Bähren 2014) Lamparth, O., Bähren, F.: From The Connected To The Autonomous Car. ATZ elektronik, Vol 9, 05/2014, Pages 36–39, Springer Vieweg Publ., 2014
- (Lu and Zhang 2007) Lu, S., Zhang, J.: Detecting unattended packages through human activity recognition and object association, PR Vol. 40, No. 8, Pages 2173–2184, 08/2007
- (Mahaffey 2015a) Mahaffey, K.: The New Assembly Line: 3 Best Practices for Building (secure) Connected Cars. Lookout Blog. August 6th 2015. Available from: <https://blog.lookout.com/tesla-research>
- (Mahaffey 2015b) Mahaffey, K.: Here Is How To Address Car Hacking Threats. TechCrunch. September 13th 2015. Available from: <https://techcrunch.com/2015/09/12/to-protect-cars-from-cyber-attacks-a-call-for-action/>
- (Markey 2015) Markey, E.J.: Tracking and Hacking: Security and Privacy Gaps Put American Drivers at Risk. 2015. Available from: https://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity%202.pdf
- (Markoff 2016) Markoff, J.: Artificial Intelligence Swarms Silicon Valley on Wings and Wheels, The New York Times online. July 17th 2016. Available from: <http://nyti.ms/2a0Awys>
- (Matheus and Königseder 2015), Matheus, K., Königseder, T.: Automotive Ethernet, Cambridge University Press, 2015
- (Maurer et al. 2015) Maurer, M., Gerdes, C. J., Lenz, B., Winner, H. (Ed): Autonomous driving, technical, legal and social aspects, Springer Vieweg Publ., 2015
- (Menn 2016) Menn, A.: Nvidia founder Huang: Artificial Intelligence triggers next Industrial Revolution (in German). Wirtschaftswoche online. 9.12.2016. Available from: <https://www.wiwo.de/technologie/digitale-welt/nvidia-gruender-huang-kuenstliche-intelligenz-loest-naechste-industrielle-revolution-aus/14951562.html>
- (Miller and Valasek 2014) Miller C., Valasek C.: A Survey of Remote Automotive Attack Surfaces. IOActive 2014. Available from: https://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
- (Miller and Valasek 2015) Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. August 10th 2015. Available from: <http://illmatics.com/Remote%20Car%20Hacking.pdf>

- (Müller and Haas 2014) Müller, M. and Haas, R.: Study on Automotive Electronics, Magility GmbH, 2014
- (Nause and Höwing 2016) Nause, M., Höwing, F.: Functional Security as a Model for Software Development in Automotive Security, ATZ elektronik, 03/2014
- (Navet and Simonot-Lion 2009) Navet, N., Simonot-Lion, F.: Automotive Embedded Systems Handbook. CRC Press, 2009
- (Ogata 2004) Ogata, K.: Discrete-Time Control Systems, Pearson Education, 2004
- (Orth et al. 2014) Orth, P., Jentges, M., Sternberg, P., Richenhagen, J.: Software Architecture and Development Tool Chain for the Drive Train, ATZ elektronik, 01/2014
- (Paar 2015) Paar, C.: The future lies in a better encryption, Interview with C. Paar, ATZ elektronik, Vol. 3, Pages 22–24, Springer Vieweg Publ., 2015
- (Pickhard et al. 2015) Pickhard, F., Emele, M., Burton, S., Wollinger, T.: New thinking for safely networked vehicles (in German). ATZ elektronik, 7/2015
- (Pickhard 2016) Pickhard, F.: Measuring everything – Big Data in Automotive Engineering (in German). ATZ elektronik, 02/2016, Volume 11, Issue 1, pp 66–66
- (Postinett 2017) AWS server down – employee shut down internet with a typo. Handelsblatt online. 2nd March 2017. Available from: <https://www.handelsblatt.com/unternehmen/handel-konsumguter/aws-serverausfall-amazon-mitarbeiter-legte-mit-tippfehler-teile-des-internets-lahm/19468246.html?ticket=ST-36653-ejHojkYxHfBvZhgrUsfa-ap2>
- (Pratap 2006) Pratap, R.: Getting Started With Matlab. 7- A Quick Introduction for Scientists and Engineers. Oxford University Press, 2006
- (Proakis and Manolakis 2007) Proakis, J. G., Manolakis, D. G.: Digital Signal Processing, Prentice-Hall, Inc., 2007
- (Reif 2014) Reif, K. (Ed.): Driving stabilization systems and driver assistance systems. Springer-Vieweg Publ., 2016
- (Rembor et al. 2009) Rembor F., Kopp, T., Herzog, S., Gugenhen, S.: Flexray – a Beginners’ Guideline, ATZ elektronik, Vol 4, 03/2009, Pages 16–21, Springer Vieweg Publ., 2009
- (Reuss et al. 2015) Reuss, H.-C., Meyer, G., Meurer, M.: Roadmap 2030 Synergies of Electromobility and Automated Driving, ATZ elektronik, 2015
- (Reuter 2015) Reuter, A.: Data security is a must for functional security (in German), ATZ elektronik, 02/2015
- (Rich and Knight 1991) Rich, E., Knight, K.: Artificial Intelligence. Mc GrawHill Publ. 1991
- (Ridder et al. 1995) Ridder, C., Munkelt, O., Kirchner, H.: Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. In: Process of Int. Conf. on recent Advances in Mechatronics. ICRAM’95, UNESCO Chair on Mechatronics, Pages 193–199, 1995
- (Russell and Norvig 2016) Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 3rd edition, 2016
- (Schaal 2012) Schaal H.-W.: IP and Ethernet in Motor Vehicles. Vector Informatik GmbH, Available from: https://assets.vector.com/cms/content/know-how/_technical-articles/Ethernet_IP_ElektronikAutomotive_201204_PressArticle_EN.pdf, Pages 1–6, 04/2012
- (Schaal and Schwedt 2013) Schaal, H-W, Schwedt, M.: New Perspectives on Remaining Bus Simulation for Networks with SOME/IP. https://assets.vector.com/cms/content/know-how/_technical-articles/IP_SomeIP_AEL_201308_PressArticle_DE.pdf
- (Schaal 2017) Schaal, S.: Auto Trends at the CES – Only four Car Manufacturers are capable of developing everything on their own. Wirtschaftswoche online. January 4th 2017. Available from: <https://www.wiwo.de/unternehmen/auto/auto-trends-auf-der-ces-nur-vier-autobauer-koennen-alles-selbst-entwickeln/19203074.html>
- (Schäfer 2010) Schäfer, W.: Software Development - Introduction for the Most Demanding (in German). Addison-Wesley Publ., 2010
- (Schill and Springer 2012) Schill, A., Springer, T.: Distributed Systems – Fundamentals and core technologies. Springer Publ., 2012

- (Seeck 2015) Seeck, A. Don't expect too much (in German), Discussion at the 1st International ATZ Conference in Frankfurt - From Driver Assistance systems to Autonomous Driving, ATZ elektronik 3/2015
- (Serio and Wollschläger 2015) Serio, G., Wollschläger, D.: Networked Automotive Defense Strategies in the Fight against Cyberattacks (in German). ATZ elektronik, 06/2015
- (Siciliano et al. 2010) Sciliano, B., Sciavicco, L., Villan, L., Oriolo, G.: Robotics – Modelling, Planning and Control, Springer Publ., 2010
- (Siebenpfeiffer 2014) Siebenpfeiffer, W. (Ed.): Networked Automobile – Safety, Car IT, Concepts (in German). Springer Publ., 2014
- (Silberschatz et al. 2010) Silberschatz, A., Galvin, P., Gagne, G.: Applied Operating System Concepts, Wiley Publ., 2010
- (Singer and Friedman 2014) Singer P.W., Friedman, A.: Cybersecurity and Cyberwar: What Everyone Needs to Know, Oxford University Press, 2014
- (Soja 2015) Soja, R.: Security and the Connected Car: Secure Networks for V2X, NXP Community Online. Available from: <https://community.nxp.com/docs/DOC-105879>, 2015
- (Solon 2015) Solon, O.: From Car-Jacking to Car-Hacking: How Vehicles Became Targets For Cybercriminals. Bloomberg online. August 4th 2015. Available from: <https://www.bloomberg.com/news/articles/2015-08-04/hackers-force-carmakers-to-boost-security-for-driverless-era>
- (Sorge 2017) Sorge, N.-V.: Top Engineer from Apple should put Tesla's AutoPilot on track. Manager Magazin online. 11th January 2017. Available from: <http://www.manager-magazin.de/unternehmen/autoindustrie/tesla-apple-topingenieur-soll-autopilot-retten-a-1129485.html>
- (Steinmüller 2008) Steinmüller, J: Image analysis – From image processing to spatial interpretation of images, Springer Publ., 2008
- (Streichert and Traub 2012) Streichert, T., Traub, M.: Electric/Electronics Architectures in Automobiles (in German). Springer, Publ., 2012
- (Sushravya 2016) Sushravya, G.M.: Cybersecurity risks in Advanced Driver Assistance Systems, Class paper, Car IT and Cybersecurity class, IIT-B, 2016
- (Thiele et al. 2013) Thiele, D., Ernst, R., Diemer, J., Richter, K.: Cooperating On Real-Time Capable Ethernet Architecture. In: Vehicles, ATZ elektronik 05/2013, Pages 40–44, Springer Publ., 2013
- (Tanenbaum and Bos 2015) Tanenbaum, A. S., Bos, H.: Modern Operating Systems. 4th edition, Pearson Publ., 2015
- (Tanenbaum and Van Steen 2017) Tanenbaum, A. S., Van Steen, M.: Distributed Systems Principles and Paradigms. 3rd edition, Pearson Publ., 2017
- (Vahid and Givargis 2001) Vahid, F., Givargis, T.: Embedded System Design, A Unified Hardware/Software Introduction, Wiley and Sons Publ., 2003
- (Vembo 2016) Vembo, D.: Connected Cars – Architecture, Challenges and Way Forward. Whitepaper Sasken Communication Technologies Pvt. Ltd. 2016. Available from: https://www.sasken.com/sites/default/files/files/white_paper/Sasken-Whitepaper-Connected%20Cars%20Challenges.pdf
- (Vivekanandan et al. 2013) Vivekanandan, B., Bavishi, H., Paranjpe, K.: Preventing malfunctions. In: E/E systems, ATZ extra, Pages 72–74, 10/2013
- (Wagner 2015) Wagner M.A.: An adaptive Software and System Architecture for Driver Assistance Systems applied to truck and trailer combinations. Ph.D. thesis, University of Koblenz-Landau, Available from: https://www.researchgate.net/profile/Marco_Wagner2/publication/279528442_An_adaptive_software_and_system_architecture_for_driver_assistance_systems_applied_to_truck_and_trailer_combinations/links/55a4f8eb08aef604aa04123f/An-adaptive-software-and-system-architecture-for-driver-assistance-systems-applied-to-truck-and-trailer-combinations.pdf, 2015
- (Weber 2013) Weber, M.: AUTOSAR learns Ethernet. Vector Informatik GmbH. Available from: https://assets.vector.com/cms/content/know-how/_technical-articles/IP_AUTOSAR_Hanser_Automotive_201311_PressArticle_EN.pdf

- (Weber 2015) Weber, M.: New Communication Paradigms in Automotive Networking. Vector Informatik GmbH. Available from: https://assets.vector.com/cms/content/know-how/_technical-articles/Ethernet_CANFD_AutomobilElektronik_201508_PressArticle_long_EN.pdf
- (Weiß et al. 2016) Weiß, G., Schleiß, P., Drabek, C.: Fail-operational E/E Architecture for Highly-automated Driving Functions. ATZ elektronik, Vol 11, 03/2016, Pages 16–21, Springer Vieweg Publ., 2016
- (Weimerskirch 2016) Weimerskirch, A.: Cybersecurity for Networked and Automated Vehicles (in German). ATZ elektronik, 03/2016
- (Winner et al. 2009) Winner, H., Hakuli, S., Lotz, F., Singer, C. (Eds.): Handbook Driver Assistance Systems (in German). Springer Vieweg Publ., 2015
- (Wolfsthal and Serio 2015) Wolfsthal, Y., Serio, G.: Made in IBM Labs: Solution for Detecting Cyber Intrusion to Connected Vehicles, Part I. Available from: <https://securityintelligence.com/made-in-ibm-labs-solution-for-detecting-cyber-intrusions-to-connected-vehicles-part-i/>
- (Wolf et al. 2015) Wolf, J., Metzker, E., Happel, A.: Ethernet-Security – example SOME/IP. Vector Informatik GmbH (in German). Available from: https://assets.vector.com/cms/content/know-how/automotive-cyber-security/Ethernet-Security_SOMEIP_Lecture_VDI_2015.pdf, 2015
- (Zetter 2015) Zetter, K.: Researchers Hacked A Model S, But Tesla’s Already Released A Patch. Wired online. August 6th 2015. Available from: <https://www.wired.com/2015/08/researchers-hacked-model-s-teslas-already/>

Links

2014

- (URL1 2014) Me, my car, my life, KPMG Automotive, 2014, <http://www.kpmg.com/Ca/en/IssuesAndInsights/ArticlesPmy-life-my-car.pdf>

2015

- (URL1 2015) <https://www.mcafee.com/de/resources/white-papers/wpautomotive-security.pdf>
- (URL2 2015) http://www.wiwo.de/unternehmen/auto/emobility/digitalisierung-der-aucht-man-das-lenkrad-nicht-mehr/v_detail_tab_print/11602152.html, 07.04.2015
- (URL3 2015) <https://www.bosch-presse.de/pressportal/de/en/bosch-and-daimler-automate-parking-mercedes-with-built-in-valet-42989.html>
- (URL4 2015) <https://www.symantec.com/content/dam/symantec/docs/white-papers/building-comprehensive-security-into-cars-en.pdf>
- (URL5 2015) <https://www.congress.gov/bill/114th-congress/senate-bill/1806/all-info>
- (URL6 2015) <https://www.digitaltrends.com/cars/bmw-automated-parking-technology-ces-2015/>
- (URL7 2015) <https://www.theiet.org/sectors/transport/documents/automotive-cs.cfm>
- (URL8 2015) https://delivering-tomorrow.de/wp-content/uploads/2015/08/dhl_self_driving_vehicles.pdf
- (URL9 2015) https://vector.com/portal/medien/solutions_for/Security/Ethernet-Security_SOMEIP_Lecture_VDI_2015.pdf
- (URL10 2015) <https://roscon.ros.org/2015/presentations/ROSCon-Automated-Driving.pdf>
- (URL11 2015) https://www.iaa.de/fileadmin/user_upload/2015/english/downloads/press/Automation-from-Driver_Assistance_Systems_to_Automated_Driving.pdf
- (URL12 2015) <http://www.team-bhp.com/forum/car-entertainment/159729-bmw-idrive-connected-drive-bmw-apps-review-faq-thread.html>

- (URL13 2015) http://www.euro.who.int/__data/assets/pdf_file/0006/293082/European-facts-Global-Status-Report-road-safety-en.pdf?ua=1
- (URL14 2015) <https://www.elektrobit.com/newsroom/webinar-automotive-ethernet-new-generation-ecu-communication/>

2016

- (URL1 2016) https://en.wikipedia.org/wiki/Advanced_driver_assistance_systems
- (URL2 2016) https://en.wikipedia.org/wiki/Safety_integrity_level
- (URL3 2016) <https://www.dhs.gov/science-and-technology/cyber-security-division>
- (URL4 2016) https://en.wikipedia.org/wiki/Functional_safety
- (URL5 2016) <http://www.exida.com/Resources/Term/Automotive-Safety-Integrity-Level-ASIL>
- (URL6 2016) https://en.wikipedia.org/wiki/Failure_mode_and_effects_anal
- (URL7 2016) <http://ec.europa.eu/programmes/horizon2020/>
- (URL8 2016) <https://www.abiresearch.com/market-research/product/1022093-connected-vehicle-cloud-platforms/>
- (URL9 2016) <http://www.acea.be/publications/article/strategy-paper-on-connectivity>
- (URL10 2016) <https://www.popsci.com/googles-cars-will-be-treated-like-human-drivers>
- (URL11 2016) <http://some-ip.com>
- (URL12 2016) <https://www.elektroniknet.de/fit-for-the-turning-point-in-the-automotive-industry-127725.html>
- (URL13 2016) <https://techcrunch.com/2016/08/25/2017-audi-a4-driver-assistance/>
- (URL14 2016) https://www.itskritis.de/_uploads/5/8/4/584ab66449001/idsposter.pdf
- (URL15 2016) <http://www.handelsblatt.com/unternehmen/industrie/elektroautopionier-tesla-ruestet-autos-zum-selbstfahren-auf/14713452.html>
- (URL16 2016) https://d23rjziej2pu9i.cloudfront.net/wp-content/uploads/2016/02/19120114/Secure_Ethernet_Communication_for_Autonomous_Driving.pdf

2017

- (URL1 2017) www.mobileye.com/technology/applications
- (URL2 2017) www.mathworks.com
- (URL3 2017) www.google.com
- (URL4 2017) <http://www.ficosa.com>
- (URL5 2017) https://www.bcgperspectives.com/content/articles/automotive-consumer-insight-revolution-drivers-seat-road-autonomous-vehicles/?chapter=4#chapter4_section4
- (URL6 2017) https://www.bcgperspectives.com/content/articles/automotive-consumer-insight-revolution-drivers-seat-road-autonomous-vehicles/?chapter=4#chapter4_section2
- (URL7 2017) <https://www.intel.com/content/dam/www/public/us/en/documents/corporate-information/policy-iiot-automotive-transportation.pdf>
- (URL8 2017) <https://www.tomtom.com/>
- (URL9 2017) <https://www.bosch.com/>
- (URL10 2017) <https://www.bosch-iiot-suite.com/>
- (URL11 2017) <https://archiv2017.iaa.de>
- (URL12 2017) <https://www.iaa.de>
- (URL13 2017) <http://www.emobil-sw.de/en/activities-en/current-projects/project-details/autoples-automated-parking-and-charging-of-electric-vehicle-systems.html>
- (URL14 2017) [https://en.wikipedia.org/wiki/Here_\(company\)](https://en.wikipedia.org/wiki/Here_(company))
- (URL15 2017) <https://www.bmvi.de/DE/Themen/Digitales/Digitale-Testfelder/Digitale-Testfelder.html>

- (URL16 2017) <https://www.vda.de/en/topics/innovation-and-technology/network/networked-mobility.html>
- (URL17 2017) https://en.wikipedia.org/wiki/Vienna_Convention_on_Road_Traffic
- (URL18 2017) https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/12837-workshop_on_governance_of_automated_vehicles_03062017_final_version-tag.pdf
- (URL19 2017) <http://adtf.omg.org>
- (URL20 2017) <https://www.rti.com/products/dds>
- (URL21 2017) <https://www.vda.de/en/topics/innovation-and-technology/automated-driving/automated-driving.html>
- (URL22 2017) <https://www.conti-engineering.com/CMSPages/GetFile.aspx?guid=c3af2186-8330-4c66-bdbd-c082502ca609>
- (URL23 2017) <https://www.kpit.com/resources/downloads/kpit-autosar-handbook.pdf>
- (URL24 2017) <https://info.glass.com/mercedes-using-adas/>
- (URL25 2017) <https://www.mercedes-benz.com/en/mercedes-benz/innovation/mercedes-benz-intelligent-drive/>
- (URL26 2017) https://vector.com/vi_security_solutions_en.html
- (URL27 2017) [https://en.wikipedia.org/wiki/Dilation_\(morphology\)](https://en.wikipedia.org/wiki/Dilation_(morphology))
- (URL28 2017) [https://en.wikipedia.org/wiki/Erosion_\(morphology\)](https://en.wikipedia.org/wiki/Erosion_(morphology))
- (URL29 2017) <http://www.pedbikeinfo.org>
- (URL30 2017) <http://www.npr.org/2017/03/30/522085503/2016-saw-a-record-increase-in-pedestrian-deaths>
- (URL31 2017) <https://www.mathworks.com/help/images/functionlist.html>
- (URL32 2017) <https://de.mathworks.com/help/vision/object-tracking-1.html>