Chapter 2

Related Work

This chapter discusses relevant work for the implementation of a reactive instinctive behavior, while considering all three aspects of an artificial intelligence agent. Therefore, the accordance of the state of the art with our proposed characteristics is evaluated considering hardware, sensors and algorithms.

2.1 Review of Autonomous Driving

The following section gives an overview of the history of autonomous driving. It proposes a separation into four main periods, each characterized by special approaches, tasks or events:

1. Mobile Robots and first steps toward autonomous driving (1969-1987)
2. Autonomous Driving on Highways (1987-2003)
3. Autonomous Off-road and Urban Driving - The DARPA Challenges (2004-2007)
4. Commercial Autonomous Driving (2007-present)

It is not possible to name all projects or developments of each period. Hence, only a few representative or outstanding examples will be given.

2.1.1 The Origin of Autonomous Driving and the First Steps (1969-1987)

An important origin of autonomous driving can be found in the field of autonomous mobile robots, where the first results were achieved over a decade before the first steps with autonomous vehicles were made.

A. Schaub, *Robust Perception from Optical Sensors for Reactive Behaviors in Autonomous Robotic Vehicles*, DOI 10.1007/978-3-658-19087-3_2

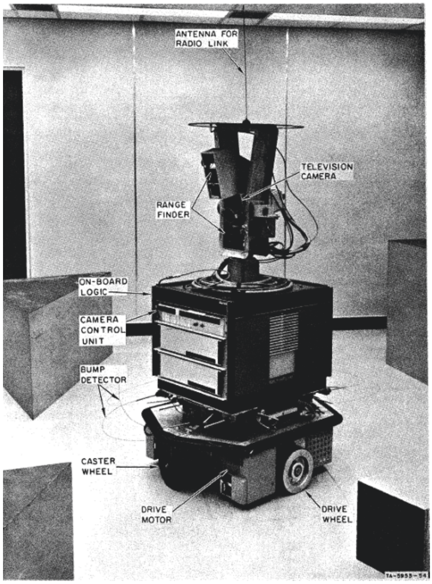

Figure 2.1: Shakey the Robot [230]

The First Autonomous Mobile Robots: Although different other machines might be classified as mobile robots, since they moved autonomously - like the tortoise robots of William Grey Walter [300], 'Shakey' was the first autonomous mobile robot that made use of the perception, planning and action principle - see Figure 2.1. The construction started in 1966 at Stanford University, whereas the first results of the complete system were achieved and published in 1969 [230]. It is remarkable that it already used vision (besides tactile 'cat whisker' sensors) for the environmental perception and motion control. Moreover, techniques were developed in the context of this project like e.g. the A* search algorithm [152] or the generalized Hough transform [109], which are still in the basic repertoire of robotics. Nevertheless, the computational power was so limited that it needed several minutes to process images and plan its path before it made a 'shaky' move.

In 1971, Hans Moravec started his work on visual navigation with the Stanford Cart (Figure 2.2), which was originally developed in the 1960s to evaluate the possibilities of remotely controlling a rover on the moon [15]. He used stereo cameras to detect obstacles and to estimate its own motion [221]. Despite the advancing computational power of micro-controllers the performance of obstacle recognition was still very slow and weak. However, the basic concepts of stereo algorithms and path finding are still used today, while the dynamic model of the robot was comparable to a car with only two degrees of freedom.

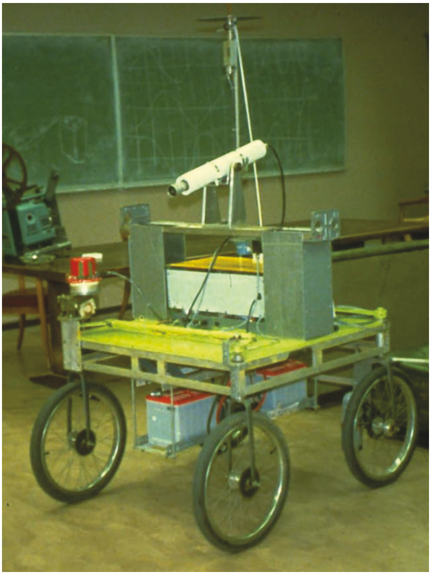

Figure 2.2: The Stanford Cart [313]

An outstanding result was achieved in 1979, when the cart successfully passed a chair-filled room completely autonomously with a speed of ≈ 0.15 $[cm/s]$ [222]
Encouraged by this work Moravec continued working on autonomous mobile robots. Now being employed at the Carnegie Mellon University he started the CMU Rover (Figure 2.3) project in 1981 [224]. Having learned from the Stanford Cart, the rover was equipped with not only cameras but also infrared short term sensors and sonar sensors for the long range. The dynamic model of the CMU Rover was more flexible with three degrees of freedom achieved by three independently steerable omni-directional wheels. Those three robots were only a few examples among many other projects that pioneered in the field of autonomous mobile robots in the 70s and 80s - see also e.g. [80]. Due to the limited processing power at that time, an extensive data accumulation was not possible and a 'fast' computation combined with an efficient representation was essential. Hence, those approaches used concepts similar to those of reactive algorithms and can be considered as their forerunners.
The first vehicle using the perception-to-action principle appeared in the mid 70s, when the Tsukuba Mechanical Engineering Lab in Japan developed an autonomous vehicle called "The Intelligent Vehicle" [286]. It used stereo TV cameras and analogue processing hardware to follow way points and to detect obstacles with a maximal driving speed of 10 $[km/h]$.

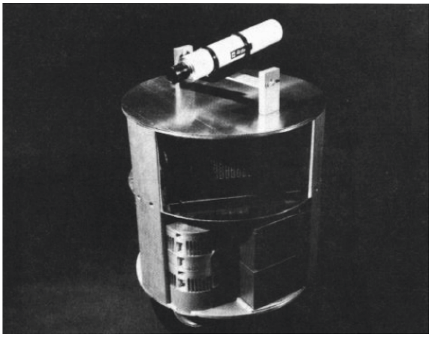

Figure 2.3: The CMU Rover [223]

Figure 2.4: The Navlab I Van [291]

CMU's Navlab: The CMU Robotics Institute did not only build up different autonomous mobile robots at that time, its Navigation Laboratory also started in 1984 the research on automated vehicles. Their first automated vehicle, the Navlab Chevrolet van - see Figure 2.4, was built in 1986 [106], but the first results were only achieved at the end of the decade due to computational limitations. The Navlab van managed to drive autonomously with a maximum speed of up to 20 [mp/h] on normal roads [279].
The earlier experiences with autonomous mobile robots definitely payed off for this project. Many algorithms developed for the autonomous vehicle could be applied to mobile robots and vice versa. For example the famous potential field approach [176], which has a reactive character (see Section 2.3.1), was an essential part of the integrated path planning and dynamic steering control [180] and the dynamic obstacle avoidance used for Navlab was developed first for mobile robots [181, 117]. At that time the Navlab van was the

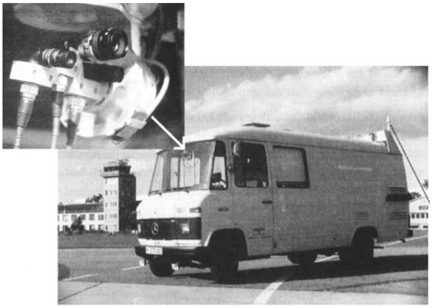

Figure 2.5: The VaMoRs Van with its Front Cameras [98]

state of the art for autonomous vehicle in the USA, but autonomous driving was also pursued in Europe.

Dickmanns' VaMoRs: At the Bundeswehr University of Munich the group of E.D. Dickmanns started to make specific hardware developments in the early 80s to enable autonomous driving. The limiting factor at that time was, same as for the mobile robots, the computational power especially for sensor processing.
A breakthrough was the BVV2 pre-processor system for image processing [143], which provided abstract features that were relevant for motion control in a hierarchical processing structure to a master PC that was in charge of the motion control. This system did not consist of a single processor but split the task to several processor (10 Intel 8086) and only certain regions selected by several search windows (32x32 pixels) were processed to be real time capable. This approach was able to extract necessary information within 20 $[ms]$, whereas standard processor were so slow that they needed around 100 $[s]$ to process an entire image.
The developments resulted in the construction of the van VaMoRs (Versuchsfahrzeug für autonome Mobilität und Rechnersehen) in 1985 [102], which was a test platform for the perception algorithms and for autonomous driving in general - see Figure 2.5. In order to be able to drive autonomously, the used approach needed a well structured highway environment with good lane-markings. The image processing hard- and software worked contour based and tried to match with high order world models (and later second order dynamic models for objects). Kalman filters were explicitly designed to consider the non-linear perspective projection. Moreover, the road curvature was determined from the visual input and served as visual feedback for the control system.
This vision-based approach was called 4D-Vision, as 3-D space was an essential part of the model as well as temporal constraints - time as fourth dimension [99].
Although the assumptions, models and techniques seem nowadays to be very restrictive, two remarkable 'world's first' achievements were made [102]. First in 1986, VaMoRs drove fully autonomously (up to 10 $[m/s]$) on a skidpan at the Daimler Benz AG in Stuttgart,

where the longitudinal and lateral motion were controlled by the output of an edge-based image processing algorithm. The second achievement was in 1987, when VaMors demonstrated autonomous high-speed driving on the closed Autobahn from Munich to Dingolfing with a maximum speed of 96 $[km/h]$ and covered more than 20 $[km]$ autonomously.
This pioneering work attracted world wide attention and demonstrated the possibility of autonomous driving.

2.1.2 Autonomous Driving on Highways (1987-2003)

In 1987 a EUREKA project called PROMETHEUS (PROgraMme for a European Traffic of Highest Efficiency and Unprecedented Safety) was initialized with a duration of 96 months until 1995 [115]. The aim of the project was to develop new concepts and solutions for more efficient and safer road traffic systems. This included infrastructure as well as vehicle developments by using (for that times) new microelectronics, information processing and aritificial intelligence. With an investment volume of 749 Million Euros PROMETHEUS is to the author's best knowledge the largest publicly funded project for autonomous driving so far. The participants were car manufactures, universities, research institutes, eletronic companies and suppliers from all over Europe. Among them were also the Bundeswehr University of Munich with Ernst Dickmanns and the Daimler Benz AG, building on the know-how accumulated in previous projects (see Section).
In 1987, the original plan of the PROMETHEUS project was to experiment with in-road cables for the vehicle guidance, but, due to the encouraging results of Dickmanns', computer vision was used instead [98] and became a prominent role of the project. In the course of PROMETHEUS Daimler-Benz presented the Vita I [288] in 1991, which was a sister-vehicle to VaMoRs equipped with its second generation of sensor and computing hardware.
Due to new algorithm in 1993, the Vita I and the updated VaMoRs were able to estimate the number of lanes and the ego-state and to identify obstacles. The authorities even allowed the group of Dickmanns to test on public roads with normal traffic [100].
In 1992, 6 years after the start of VaMoRs, a new vehicle called VaMoRs-P or VaMP was introduced [101]. The computers were small enough, so that a Mercedes 500SEL passenger car could be used instead of a 5-ton van. The twin vehicle at Daimler-Benz was the Vita II [289], which only had slightly different perception system. Two spectacular demonstrations were given with the new sedan cars that attracted worldwide attention. In 1994 at the final presentation of the PROMETHEUS project VaMP drove in real traffic around Paris with up to 130 $[km/h]$, while automatically changing lanes and overtaking slow vehicles [101].
After that the computer hardware was renewed with a magnitude faster PowerPCs and in 1995 VaMP managed to drive 95% of the 1600 $[km]$ long way from Munich to Odense, Denmark autonomously [103]. The maximum speed reached while being in automatic mode was 180 $[km/h]$. Both longitudinal and lateral control was done again by the artificial intelligence, whereas at the CMU's 'No Hands Across America' demonstration the NavLab 5 vehicle was automatically steered for 98% of the time, but a human operator had to control the longitudinal speed [243].

Figure 2.6: The VaMoRs-P 500SEL [102]

Figure 2.7: The Lancia Thema 2000 from the ARGO Project [50]

After the PROMETHEUS project ended the research on autonomous driving on highways continued - for instance in the ARGO project of the Artificial Vision and Intelligent Systems Lab (VisLab) of the University of Parma, which also took part in the PROMETHEUS project. The ARGO vehicle, see Figure 2.7, drove in 1998 the 'MilliMilia in Automatico', which was a journey through Italy, where 94% of the 2000 [km] were covered autonomously [50].

With the experience of over 10 years of autonomous driving the group of Dickmanns extended its research to other scenarios. The old VaMors van was still used from 1994 until 2004 with the cameras and the processing hardware being revised a third time [98]. The goal then was to drive in more unstructured environments and even off-roads. In 2004 VaMoRs demonstrated in an autonomously executed mission its capability to leave the road to go off-road, to avoid even negative obstacles (ditches) and to reach an off-road target position [236, 98].

2.1.3 The DARPA Challenges (2004-2007)

The advances of autonomous vehicles in the new millennium were mainly driven by competitions hosted by the American Defense Advanced Research Projects Agency (DARPA) and the European Landrobot Trial (ELROB). Well structured environments like highways were left and off-road or urban scenarios provided new challenges.

Figure 2.8: The Winner of the DARPA Grand Challenge 2005: 1. Stanley, 2. Sandstorm, 3. H1ghlander [59]

DARPA Grand Challenge 2004: Nearly ten years after the PROMOTHEUS project was finished the DARPA announced the Grand Challenge to promote research on autonomous ground vehicle. In this challenge, an autonomous mobile robot had to navigate through an unknown territory. The first Grand challenge was on March 13 in 2004 in the Mojave desert from Barstow, California, to Primm, Nevada. The team, whose robot would have covered the 142-mile track fastest and in no more than 10 hours, would have been rewarded with 1 million dollars.
Even though 106 teams were registered and 15 participated in the final race, the Red Team from Carnegie Mellon University, performing best, could cover only 5% of the course [59]. Its autonomous vehicle called Sandstorm (see Figure 2.8) was counted as the favorite since it qualified first and was the only autonomous vehicle that could comply with all the pre-event rules. However, due to weaknesses in the perception, navigation and planning modules, it collided with several fence post and was finally stopped by a large boulder stone [295]. Nevertheless, Sandstorm traveled 7.4 miles with an average speed of 15 [*mph*] and a top speed of 36 [*mph*]. The main issues arose because of algorithmic errors and not because of limiting computational power. However, valuable experiences regarding cross-country autonomous driving were made during this Grand Challenge. At the closing ceremony a next edition of the DARPA Grand Challenge was announced for October 2005.

DARPA Grand Challenge 2005: On October 8 in 2005, only 18 month after the first Grand Challenge, the second edition was hold with a 2 million dollars reward for the winner. Only 23 teams out of 197 applicants were selected in the qualification process to participate in the final race. The finalists had to prove that they are able to perform better than all vehicles of the Challenge of the preceding year. Again a difficult 132 mile route was chosen which led through the desert roads of Nevada. The 2005 route was slightly less demanding since no obstacles were directly placed on the roads and precise GPS data was provided [59].
In contrast to the first Challenge 5 teams finished the course successfully and 4 even within the 10 hours time-limit. The winner was the robot Stanley from the Stanford University with 6 hours 53 minutes. The 2nd place took Sandstorm from CMU's Red Team, which also participated in 2004, with 7 hours 5 minutes. Rank 3 was H1ghlander with 7 hours 14

minutes, which is a twin robot of Sandstorm and also from Red Team. The race times of the three winning vehicles were all well below the 10 hours and were very similar, although the chosen hardware (compare Figure 2.8) and especially the implemented algorithms had significant differences.
Stanley was based on a VW Touareg and used mainly off-the-shelf hardware like e.g. its laser scanners, which where the main perception sensors together with a camera system. The main focus was laid on the development of artificial intelligence algorithms and especially in the fields of machine learning and probabilistic reasoning [282].
The CMU's robots Sandstorm and H1ghlander were both constructed based on a 998 HMMWV and Hummer H1 in accordance to the team's strategy of keeping components as simple as possible and thus robust [294]. Although both robots have different electro-mechanical hardware like e.g. the steering actuators, they use the same combination of LIDAR and RADAR for environment perception and the same navigation algorithm. Nevertheless, both were run with different parameter settings during the race, H1ghlander with a full speed strategy and Sandstorm with a more conservative one, in order to maximize the probability of at least one robot finishing. Despite the conservative strategy Sandstorm finished the race faster, as H1ghlander encountered technical problems during the race.

DARPA Urban Challenge 2007: In 2007 the next Darpa Challenge was carried out, but this time on a totally new level. The scenario was moved from the static and easily model-able desert terrain to the more challenging dynamic urban environment of an entire mock-up town belonging to the US Army. The task this time was much more complex, since it was not a conventional race as in the Grand Challenges. The robots had to fulfill a series of different missions, which were defined by a set of check points.
After the announcement 89 teams coming mainly from universities with industry partners applied and submitted first a technical paper, which had to sketch the respective concepts of an autonomous urban vehicle. This led to the invitation of 53 teams from all over the world, which presented their vehicles and demonstrated the capability of basic autonomous driving maneuvers. In the next round, the remaining 36 teams had to accomplished a difficult qualifying event with several stages to proof the capability of following way points, evading obstacles and obeying the traffic rules to ensure certain quality and safety standards. Nevertheless, 11 teams got the admission for the 97 $[km]$ long final race on November 3. The missions in the final event included parking in a defined parking lot, overtaking parked or slow vehicles, executing U-turns in case of blocked roads, merging into fast-moving traffic, turning left while observing the traffic on the other lane, and handling the right of way at intersections with multiple stop signs. During the whole time the vehicles had to move within traffic that consisted of the other robots participating in the final and additionally 50 especially trained human drivers. The speed limit was 48 $[km/h]$ on the fastest parts of the course. Despite those challenging conditions three vehicles finished the challenge without human intervention and three other teams could solve all of the given tasks with only small interventions.
The winner of the urban challenge was Team Tartan Racing from Carnegie Mellon University with its vehicle Boss, a 2007 Chevy Tahoe [293] with drive-by-wire modifications.

Figure 2.9: The Winner of the DARPA Urban Challenge 2007: 1. Boss [293], 2.Junior [220]

Figure 2.10: Left: Team AnnieWAY [171], Middle: TerraMax [74], Right: MuCAR3 at ELROB 2010 [158]

Although the team could gather experiences in the preceding DARPA Challenges, the totally different tasks required a lot of new developments like a planner for unstructured as well as structured environments, a tracker for moving obstacles, and a mission and behavioral planner. The sensory equipment consisted of different LIDARs, radar, only one camera and an IMU aided GPS module, which became crucial in the narrow urban environment. Boss performed best already in the qualifying and, therefore, could start from the pole position to finish the course with the fasted time, which was 4 hours 10 minutes with an average speed of 22.5 $[km/h]$. Ranked second place was a VW Passat called Junior, which belonged to Sebastian Thrun's Stanford team that won the Grand Challenge 2005 [220]. It used a modular software architecture based on the one used in 2005 with revised and new modules for sensor interfaces, perception, navigation, drive-by-wire, and global services. The planning part had also to be modified to enable a global mission planning together with one planner for driving on roads, one for unstructured environments, and a state chart realizing the behavioral planer to initiate actions like U-turns. Junior used 5 different types of laser measurement systems, a multi-radar assembly and also an IMU-aided GPS. Although the pure operational race time was 4h 5min, Junior was ranked 2nd with a time of 4h 29min due to being stuck behind two vehicles that crashed. The average speed of 22 $[km/h]$ was lower than the value of the Grand Challenge in 2005, but this is caused by the speed limits and the complicated environment.

Several European teams also managed to reached the final like Team Carolo from Technische Universität Braunschweig with its vehicle Caroline [247], which couldn't finish the course, and Team AnnieWAY from Karlsruhe Institute of Technology in cooperation with the former institute of Dickmann from University of the Bundeswehr [171], who didn't complete the race either. Both teams and Stanford used a VW Passat, a popular choice at that time because of an available kit for autonomous driving making CAN bus partially accessible and enabling electric steering, braking, accelerating and gear shifting.

Among the finalists was also a team from Vislab of the University of Parma with an Oshkosh MTVR Truck called TerraMax. This truck is originally a military vehicle and therefore not a typical urban vehicle [74]. A remarkable point was also the sensory equipment as they relied mainly on 11 cameras [51], which were used in mono, stereo and trinocular setups to cover the environment. TerraMax had to quit the race after 90 minutes due to a computer error, whereas it should be mentioned that all finalists had problems. Even the winning teams encountered different minor bugs either due to logic, hardware or GPS errors.

Many lessons could be learned from the urban challenge such as no available off the shelf sensors was sufficient - even not the 64 beam Velodyne laser [293]. Moreover, the used models were too simple for urban driving, as no pedestrians, traffic lights and bikes were considered. Very often traffic jams had to be solved by humans and often operators had to intervene. The urban challenge also demonstrated that it is not possible to consider each scenario that may occur in a realistic traffic environment. Hence, this motivates a reactive instinctive behavior, as a high abstraction ability and adaptivity are required to establish safety modules being able to cope with un-modeled scenarios. Nevertheless, the urban challenge promoted various innovations in the fields of moving obstacles detection, localization, mixed mode planning, curb detection, vehicle tracking, and the behavioral planners that cope with a broad range of traffic situations. This technology demonstration attracted a lot of international attention and made huge impact on car manufactures, so that they intensified again their research on autonomous driving. However, the development of reactive approaches was of minor importance in this period, as the focus laid on accumulating and fusing large amount of data from different sensors (see also Section 2.2.2).

European Landrobot Trial (ELROB): Besides the three DARPA challenges a series of competitions for autonomous vehicles was established in Europe in 2006 called ELROB (European Landrobot Trial). ELROB is part of European Robotics, which is a non-profit organization with the goal to bring together users, researchers and industry to promote European robotics [111]. The ELROB wants to clearly distinguish itself from being a competition similar to the DARPA Challenges and defines itself as a demonstration of the current capabilities of European robotics.

The first trial was organized by the German Army, but in 2007 an additional civilian track was established. The focus at the trials lies on real world scenarios, whereas no simplifications like detailed maps or well visible road markings are provided. The missions take place on- and off-roads and include different categories like transport convoy, camp security, transport mule, reconnaissance and surveillance, and autonomous navigation. The latter two categories are of special interest, as they demand a high level of navigation in unstructured environments. Due to this fact and the lack of a stable GNSS connection

Figure 2.11: Left: Google's Prius in 2011 [145], Right: Google's Lexus RX450h in 2012[140]

or a reliable map, reactive approaches and concepts are used for the autonomous control of the ELROB vehicles (like in [158, 299]). The ELROB trials were held every year between 2006 until 2012 with alternating military and civilian version. Since 2012 the trial takes place every two years with a variety of scenarios in urban and off-road environments.

2.1.4 Commercial Autonomous Driving (2008-present)

After the DARPA Urban Challenge demonstrated the technical possibilities, the dream of driving autonomously got back into consciousness worldwide. The major car companies started to intensify their research on advanced driver assistance systems and first dates were estimated, when an autonomous vehicle could be bought. GM plans to deliver Cadillacs that are able to drive automatically in uncritical situations in 2017 [203] and Nissan announced in 2014 even a fully autonomous vehicle for 2020 [26]. Besides the technical topics there is another big question mark regarding insurances [168] and especially legal issues. Therefore, politics started to give an increased attention to autonomous driving [131]. In 2011 Nevada and California were the first states to legalize autonomous driving on public roads [214, 242], while in 2013 Florida, Michigan and District of Columbia followed their example. The current status of all states can be checked in [120]. European manufacturers shouldn't have a disadvantage, so that politics allowed autonomously driving on certified roads in England [33], France and Germany [248, 285].
The development of technology for autonomous vehicles attracted not only classical car manufacturers. The best example here is Google, which started already as a sponsor of the Standford teams in the DARPA challenges in 2005 [282] and 2007 [220]. The former leaders of the Stanford team Sebastian Thrun and Chris Urmson from the Carnegie Mellon University [294, 293] became heads of the autonomous driving group at Google [145] starting in 2009 [244]. They set up a fleet of Toyota Prius and changed later to Lexus RX450h [140], which were driving autonomously first in California and then later also in Nevada. Starting officially in 2011 the fleet covered over 1.5 million kilometers autonomously until May 2015 [292] while consisting then of over 20 vehicles.

Those vehicles depended strongly on the Velodyne 64-beam laser and accurate maps [145], for which the experience that was collected for years from Google Maps and Google

Figure 2.12: Google's Self-Driving Prototype Vehicle [142]

StreetView paid off. Although to be considered as very safe, the Google cars have been involved in 11 minor accidents without any human damage until May 2015 [34], whereas Google stated that the artificial intelligence was never the cause for any of those accidents [244].
A further step was made in May 2014, when Google announced a car directly designed for autonomous driving [141]. The special shape reminded of a golf cart with a Velodyne Lidar on rooftop. Another very unusual feature is the missing steering wheel and pedals in the first prototype, so that it cannot be controlled by a human operator. Google started tests on public roads in Mountain View, CA in the summer of 2015 [142]. For those tests a speed limit of 25 [*mph*] is set and safety drivers will be aboard, who can overtake the control of the vehicle with an added steering wheel and pedals. The software basis is the same as for the autonomous vehicles that are based on conventional vehicles.

Another new player is the electric car manufacturer Tesla, who announced an autonomous driving function for its Model S for summer 2015 [13]. In fact a combination of the existing driver assistance systems like the automatic emergency braking, blind spot observation, collision warning sensors, and lane marking detection enable a hands-free driving on highways and an automatic lane change that must be triggered by the driver [253]. Additionally, Tesla released a low-speed automatic driving for parking lots. The speed had to be very limited since ultra-sound sensors are used for perception. Nevertheless, the sensory equipment is not compatible to autonomous cars like the ones from Google's fleet. A fatal accident occurred in July 2016 as a driver was relying on the so called 'autopilot' functionality of the Tesla S and crashed into a turning truck on a highway [266].

The 2014 Mercedes S-class has the capability of driving automatically on highways and well defined streets, but requires the driver to keep his hands on the steering wheel due to legal issues. If the driver doesn't comply, the car warns him and eventually forces him to drive to the side of the road and to stop [13]. Daimler has a long tradition regarding autonomous driving. All started with Dickmanns's presentation of VaMors

Figure 2.13: Audi's RS7 'Jack' [25]

at the Daimler Benz AG in Stuttgart in 1986 and the following common projects with the test vehicles Vita I [288] and Vita II [289]. With the experience gathered in the EU Projects a steering assistant was developed in 1994 [129], but image processing for driver assistance systems became a prominent topic at Daimler. The group around Uwe Franke worked on computer vision algorithms for e.g. lane recognition [312] or vehicle tracking [137] and developed early real-time stereo vision approaches[128]. The shift from highways to urban areas already started in 1998 [124], while the complex environments raised many new topics starting from bus stop recognition [127], new collision avoidance techniques [130], road modeling [302], efficient representations of the environment [240] and many more. A breakthrough was the combination of the 3D reconstruction algorithm Semiglobal Matching [159] with optical flow [227, 303] to the 6D Vision approach [245]. With this the environment could be modeled in 3D and dynamic objects could be detected with only using a pair of calibrated cameras. The 6D-Vision approach is used since 2013 in the Mercedes-Benz S-, E- and C-class to observe crossing traffic and pedestrians [125]. Moreover, in August 2013 a Mercedes Benz S500 Intelligent Drive completed the 100 kilometer long Bertha Benz route through dense traffic and complex urban and rural environments autonomously [216]. Bertha Benz, who was the wife of the patent motor car inventor Carl Benz, covered the same route exactly 125 years before and demonstrated the first long distance drive of an automotive.
Besides Mercedes, the other premium manufacturers made also remarkable achievements on their way to build autonomous vehicles. Audi proposed and demonstrated an approach for autonomous or piloted parking [166] in a parking garage, which is equipped with LIDAR sensors. Another topic tackled by Audi is high-speed autonomous driving on a race track with the vehicle being close to its handling limits [133]. The current autonomous vehicle named 'Jack' from the piloted driving project is an Audi RS7 (2.13) that already demonstrated its autonomous driving capability on the highways of Nevada and on the German Autobahn A9 near Ingolstadt [25].

Figure 2.14: The autonomous Freightliner Inspiration Truck [41]

BMW also focuses on automated driving in well structured environments like highways. Since 2011, BMW has a fleet of test vehicles that drives on the German Autobahn A9 with speeds up to 130 $[km/h]$ [16]. The autonomous driving research belongs to the 'connected drive' group [107]. This makes thematically sense, since an intelligent vehicle is closely connected with its environment and to the internet to obtain additional information about e.g. the traffic density.

Not only premium car manufactures work on autonomous driving. Volvo as company with a long tradition of vehicle and crash safety started to work on automated driving for pedestrian safety with the goal of 'zero deaths' in car accidents by 2020 [93]. In 2017 Volvo plans to launch 'Drive Me', an experiment in which 100 autonomous vehicles are provided to normal customers around Gothenburg. The testers will drive in normal traffic with a limited speed of 30 $[mph]$ during autonomous mode.

Besides passenger cars, the automation of trucks is another interesting area of research, since they cover long distances, drive relatively slowly and are involved in heavy accidents often caused by tired drivers [31]. Mercedes has presented an autonomously driving truck called Mercedes-Benz Future Truck 2025 at the IAA 2014 [40]. The same technique called 'Highway Pilot' is used in the Freightliner Inspiration Truck 2.14 that drives since 2015 autonomously through Nevada [12].

For completeness it must be mentioned that besides the large car manufacturers and technology companies research institutions still pursue independent projects in the field of autonomous driving. For example, the VisLab of the University of Parma attracted a lot of attention with the Vislab Autonomous International Challenge in 2010, in which a convoy of two electric vehicles drove partially autonomous 13000 $[km]$ from Parma to the World Expo in Shanghai [54]. The leading vehicles was driven autonomously when possible and the following vehicle tried to locate the leader and follow it. If this was not possible the second vehicle planned its way according to a provided GPS way-point list. Furthermore, in 2013 VisLab demonstrated autonomous driving in downtown Parma within the PROUD (Public ROad Urban Driverless-Car) project [52]. Undoubtedly, the ability of driving autonomously will be part of future vehicles. However, more daring

Figure 2.15: Mercedes F-015 Luxury in Motion [42]

concept cars like the Mercedes ‘F 015 Luxury in Motion’ - see Figure 2.15 - are indicating that future cars are not like today’s vehicles with just the ability to drive autonomously. This very ability enables a new usage for vehicles not as a mere transportation utility but as a “mobile living space” [42]. This prototype is also remarkable, as it is one of the first concept car that was developed fully for autonomous driving. Even though the research of autonomous driving has gained momentum since the DARPA Urban challenge, there is still long way to go for the fully autonomous vehicle, to which the concept of an artificial instinct could contribute.

Although remarkable results have been achieved, so far no vehicle is able to act autonomously in nearly all situations. Moreover, none of them is pursuing a reactive instinctive approach. Most of the vehicles are conventional series cars (the vast majority uses combustion engines) that have been adapted for autonomous driving by integrating additional sensors and actuators. However recently, a few different projects like the Mercedes F015 concept [42] were initiated to build a car explicitly with the purpose of autonomous driving, even though it is not known and rather questionable if the de-centralized systems architectures used in current vehicles was rethought. Most important, all the introduced vehicles have the motion dynamics of a conventional car, and therefore do not fulfill the requirements of a highly maneuverable vehicle. Consequently, from an actuator point of view the current autonomous vehicles are not perfectly suitable for reactive instinctive approaches.

2.2 Sensors for Autonomous Vehicle

It is not in the scope of this thesis to develop or design a new sensor for autonomous vehicles, but a sensor shall be selected that supports reactive behavior for autonomous vehicles. This section first provides a survey of common sensors for autonomous driving and then gives an overview which sensor classes were used for the different vehicles through the decades.

2.2.1 Overview of Sensor Classes

We discuss now different sensor classes that are used for autonomous driving. Even though a great variety within single sensor classes exists and major evolutions were made during the last 30 years (considering e.g. the available resolution of cameras), the basic underlying principle of a sensor class is always the same. The following sensor types are only briefly introduced, while a more detailed description can be found e.g. in [264, 226, 306].

- **Odometry**: Odometry denotes the general principle of measuring wheel speeds and integrating them to estimate the relative position change over time. The word has its origin from odos (Greek - 'route') and metron (Greek - 'measure').
- **SONAR**: The SOund Navigation And Ranging sensor actively transmits electromagnetic waves in the spectrum fro 16 $[kHz]$ to 1 $[GHz]$ and calculates the distance to reflecting objects.
- **RADAR**: The class of RAdio Detection And Ranging (RADAR) sensors transmits microwaves (1 $[GHz]$ - 300 $[GHz]$), which reflect from objects and are then detected by the receiver of the radar system to estimate distance and velocity.
- **LIDAR**: LIght Detection And Ranging (LIDAR) is closely related to RADAR systems, but uses laser beams instead of radio waves.
- **Monocular Camera**: A camera is a passive sensor system that projects the visible light (and other parts of the electromagnetic spektrum) on an electronic sensor plane to record an image of the environment.
- **Infrared Camera**: An infrared or thermographic camera is very similar to a conventional camera, but instead of visible light it detects infrared radiation. Therefore, it is able to visualize the surface temperatures of objects.
- **Stereo Camera**: A pair of cameras is able to reconstruct depth form two concurrently captured images using triangulation. Usually, such a configuration requires additional synchronization and calibration.
- **IMU**: An Inertial Measurement Unit combines an accelerometer and a gyroscopes for each spatial axis to measure accelerations and rotations.
- **GNSS**: A Global Navigation Satellite System (GNSS) like the Global Positioning System (GPS) is a network of satellites, which transmit time signals so that a receiver can determine its position on earth by receiving at least 4 satellites. Often a correction signal like the Differential GPS [219] from an earth-bound station is used to refine the position.

From now on the SONAR sensor is omitted, as its relevance for autonomous driving is minor due to the short range and resolution. Though, it is still used for automatic parking assistance systems [146]. Furthermore, infrared cameras are only rarely used, as in [257] or[231], for applications in the context of autonomous vehicles and therefore not

considered explicitly anymore.
Table 2.1 gives an overview which measurement from the list in Section 1.3 can be sensed by which system. Different measures like odor, taste, and tactile sensing are not included as being considered not practical, whereas acoustics is only left out due to its rare use in automotive applications until now. Nevertheless, sound-based localization [110] and control is used in mobile robotics [202], and a human driver also observes acoustic signals.

Sensor	Acceler-ations	Velocities	Depth	Color	Texture	Geometry	Position
Odometry	-	x	-	-	-	(x)	(x)
Radar	-	x	x	-	-	-	-
Lidar	-	x	x	-	-	-	-
Monocular Cam.	-	-	-	x	x	x	(x)
Stereo Camera	-	-	x	x	x	x	(x)
IMU	x	x	-	-	-	-	-
GNSS	-	-	x	-	-	-	x

Table 2.1: Overview of Sensors and what they could measure

It is already obvious that cameras tend to be good sensors, as a great variety of information can be acquired. Although, stereo cameras have the advantage of the depth determination, they require a high computational effort [128], additional calibration and have a limited range in context of autonomous driving.

2.2.2 Usage of Sensors

This subsection provides an overview how the different sensor types were used in exemplary autonomous vehicles of the last 30 years. Table 2.2 is in chronological order, while the border between two era's, in accordance with the preceding section, is denoted by a double horizontal line. It can be seen on the first glance that two sensor types are common to all autonomous vehicles of that table: Odometry and (monocular) cameras. This is also the minimal sensor equipment which was used e.g. for the VaMoRs vehicles. Moreover, it is also an intuitive choice, since the vehicle's control relies normally on the odometry feedback and cameras provide a rich variety of data from the environment, while roads are designed for humans relying mostly on their eyes.
Until the DARPA challenges cameras were the dominant environment sensor, but then new RADAR and LIDAR systems provided depth measurements without the high computational effort of e.g. stereo cameras. Since the autonomous vehicles left the well-structured highway environment, the environment was modeled by a combination of LIDAR and RADAR to detect near and far objects, and was fused with a global map containing GPS way-points [294]. A deciding factor for the success were the Simultaneous Localization And Mapping (SLAM) algorithms for the local environment [282] and the DGPS/IMU combinations that provide high accuracy information for global localization

Autonomous Vehicle	Odometry	Radar	Lidar	Mono. Camera	Stereo Camera	IMU	GNSS	Year
VaMoRs	x	-	-	x	-	-	-	1985
Navlab	x	-	x	x	-	-	-	1986
VaMoRs-P	x	-	-	x	-	-	-	1992
ARGO	x	-	-	x	x	-	-	1999
TerraMax	x	-	x	x	x	x	x	2004
Stanley	x	x	x	x	-	x	x	2004
Sandstorm	x	x	x	x	-	x	x	2004
Boss	x	x	x	x	-	x	x	2007
Junior	x	x	x	x	-	x	x	2007
Bertha	x	x	-	x	x	x	x	2013
Google Car	x	x	x	x	x	x	x	2014
Jack	x	x	x	x	x	x	x	2014
Future Truck	x	x	-	x	x	x	x	2014

Table 2.2: Sensor equipment of the autonomous vehicles through the decades

[47]. Besides a few exceptions like TerraMax [51], cameras were only used for detecting drivable regions [282] or lane markings [293].
Nowadays, due to dense and real-time capable 3D reconstruction algorithms like [159], stereo cameras are used in autonomous vehicles together with LIDAR and RADAR. Although, the use of different sensor types provides robustness by redundancy or complimenting properties [160], not all combinations are yet suitable for mass production due to high costs. An extreme example is the $70,000 Velodyne 64-beam LIDAR used e.g. by the Google Car [151] or Junior [220]. Automotive suppliers and manufacturers are working on solutions that are closer to series-production like in the Mercedes autonomous vehicle Bertha or in the autonomous future truck.
In conclusion, cameras have been an essential sensor for autonomous vehicles through the decades due to the variety of information they can provide and computer vision being a large field of research. After all, living creatures also navigate mainly with their eyes. Cameras are a very promising sensor for realizing a reactive behavior. This thesis is supported by the next section that gives an overview over the state of art of reactive algorithms.

2.3 Types of Planning

The core element to realize a reactive behavior is the algorithmic part that connects the sensory part with the actuator hardware. The term reactive algorithm or reactive

planning was often used for a rule-based decision making, or also called Situated Control Rules [108], in order to cope with uncertainties [259] or limited computational power, which is one of the reason why those algorithms were popular in the late 1980s and 1990s. The main field of application of those reactive planning approaches is robotics like in [35], where structured reactive plans determine the responses of a robotic system to sensor changes. Furthermore, this “reactive school” [23] was often closely related to a task-based architecture [235] as proposed by Brooks [56].
Those reactive planning algorithms are only partially in accordance with the postulated properties of a reactive instinctive behavior. The rule-based response to sensor signals is similar to the reflex definition of Chapter 1. Since a task is executed by a sequence consisting of a set of rules, this class of algorithms is a series of reflexes. The representation interfacing the sensor space to the actuator space is limited to almost binary decisions and therefore, graphs or finite state machines, which are not necessarily deterministic [169], are used to model and cause the desired behavior of the system.

Exemplary Classification of Planning Types: In the context of autonomous vehicles, [200] proposes a classification of motion planning/ navigation algorithms into global, reactive and guided navigation.
The global navigation is characterized by a trajectory planner that uses a metric representation of the environment (map). An autonomous vehicle will then track the computed trajectory [305]. This form of navigation pursues a clear hierarchical architecture, where no sensor-actuator coupling exists because of the map-building step. In contrast, the reactive navigation couples directly navigation to perception. Additionally, an object recognition ability is postulated to notice when reaching a goal state. The tentacle approach [299], where motion primitives are evaluated on an occupancy grid, is named as an example for a reactive navigation. Furthermore, a third type, namely the guided navigation that is settled in between the global and reactive navigation, is proposed. This type avoids metric planning and causes no random motion patterns, which is stated as an attribute of the reactive navigation. Platooning is given as an example application for guided navigation [213].
Figure 2.16 visualizes the different navigation types by using characteristic values of the amount of information required and cycle times. Global navigation is definitely different to a reactive instinctive behavior e.g. due to the decoupled sensor-actuator-relation, but the definition of a reactive navigation in [200] is too restrictive. Another example provided for reactive navigation is a robot that bounces off when hitting a border. This random motion is not very constructive and would not be considered as instinctive navigation. In the best case, the bouncing could be seen as a reflex. Moreover, they state that transferred to autonomous vehicles those boundaries would be e.g. curbs. On the other hand road following is given as an example for guided navigation. This leads to the question why road following is so much different to navigate in a polygon spanned by curbs etc. If their reason for this distinction is the sensor-actuator coupling, since a curb can be directly detected by the LIDAR that they use and road boundaries not in all cases, then probably the wrong sensor is used.
Nevertheless, using little additional knowledge is still in conformity with intuition and

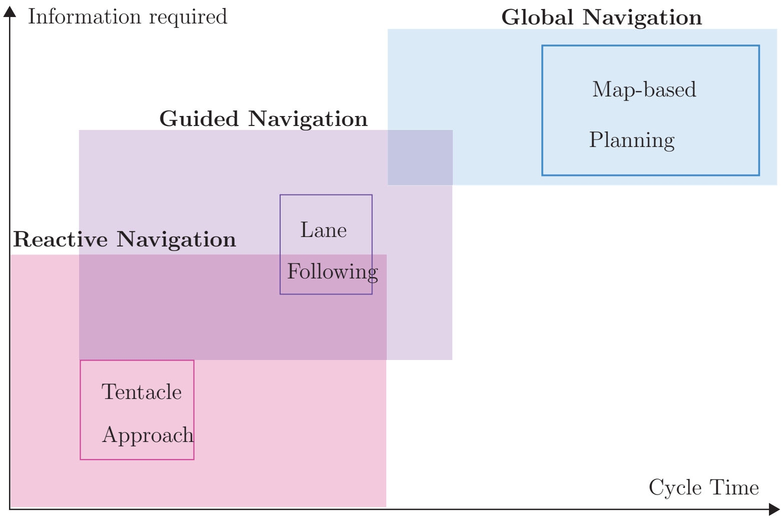

Figure 2.16: The classification of Wuensche [200] into reactive, guided, and global navigation is visualized here according to the required amount of information and the cycle time. An exemplary algorithm for each class is drawn at its respective position.

the postulations from the initial chapter. In [24] Arkin also proposes the use of "world knowledge" to extend the reactive algorithm class described at the beginning of this section. Therefore, the algorithms that realize a reactive instinctive behavior are covering both the guided and the reactive navigation.

The remainder of this section discusses algorithms that comply at least partially with the postulated properties of a reactive algorithm in the first chapter. Recalling the task-based architecture of Brooks [56], a reactive instinctive behavior is realizable on the lowest levels. Therefore, the approaches presented in this related work section are ordered according to the task they fulfill, while the sensor or the system they run on is subordinate. The following task-based structure is pursued and depicted by Figure 2.17 even more detailed:

1. Obstacle avoidance.

2. Relative position to static or moving target poses is called navigation.

3. Combined obstacle avoidance and navigation.

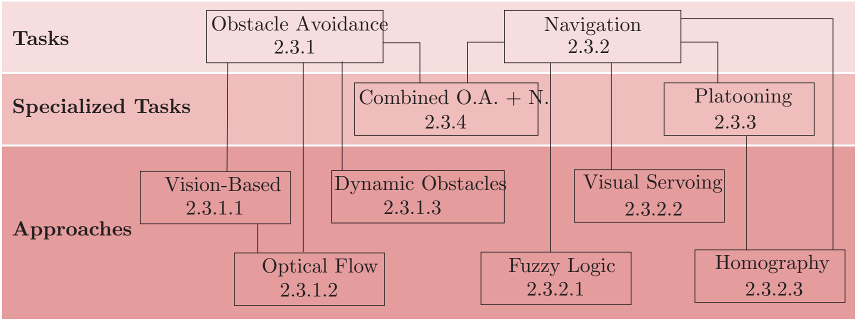

Figure 2.17: Overview of the structure of Section 2.3

2.3.1 Reactive Obstacle Avoidance

Obstacle avoidance is the most basic task for autonomous vehicles and mobile robots as seen for example in Figure 1.3. A large variety of different active and passive sensors were applied to fulfill that task during more than 40 years of mobile robotics [264].
In context of autonomous vehicles and Advanced Driver Assistance System (ADAS), this variety is limited mainly to the sensors listed in the preceding section, namely RADAR [97], LIDAR [155], stereo cameras [51, 311], and combinations of those [71].
A collision-free motion requires a fast response time and robustness for obstacle detection even in cases where the calibration of the system changes due to external factors like vibrations. Therefore, obstacle avoidance is the dominant task in literature that is solved by reactive approaches, whereas not all fulfill the postulations from the first chapter. However, the algorithms described in this section do not rely on data abstraction but try to couple the response directly to a sensor input. Moreover, it is important to detect collisions as early as possible and especially with dynamic obstacle.

Tentacle Approach: The first algorithm is the Tentacle approach introduced in [299]. An occupancy grid map is generated from LIDAR depth measurements and motion primitives that dependent on the current velocity and wheel angles are projected in this metric map to identify collision free paths. The tentacle obstacle avoidance is named as an example for a reactive algorithm in [200], but a critical point is the data accumulation. The accuracy of the occupancy map is directly dependent on the number of measurements, and different methods like [280], which uses a probabilistic approach, were developed to increase the accuracy. Therefore, there is a conflict of objectives between accuracy and preserving the reactive character. Nevertheless, this approach was successfully applied in off-road scenarios like the ELROB challenge [158] or by Team Annieway [171] in the DARPA Urban Challenge.

Physically inspired obstacle avoidance: Many reactive collision avoidance methods are inspired by physical principles like the popular potential field approach introduced

by Khatib [176]. The basic idea is that the environment is considered as a potential field, like a gravitational or electric potential, and an imaginary force is repelling the mobile robot from obstacles, whereas the goal position is attracting it. The motion of the robot is then caused by the sum of all forces. Various improvements were developed to avoid for example local minima by using harmonic potential functions [177] or circular fields [265]. However, the environment representation is often a metric one like the occupancy grid map or a similar form, so that the problem of data accumulation holds. Considering explicitly the goal position, the potential field approach is not used as reactive algorithm, but as a path planner.
The dynamic window approach [121] takes the dynamics of the robot into consideration to apply only feasible velocities. With this knowledge, an objective function is maximized to ensure a fast translational motion and keeping a safe distance to obstacles. Although the algorithmic layer is closely coupled with the hardware, the sensor properties are neglected by this algorithm.

For completeness it should be mentioned that reactive obstacle avoidance is also often used in the field of robot manipulators. Exemplary approaches are based on forces [148], settled in the Cartesian space [228], or in the operational space [49]. One important difference is that the configuration space has a significant higher dimension than for ground-based mobile robots or autonomous vehicles [48], which leads to a higher computational effort, but can also provide more feasible solutions.
There is a large number of different approaches in literature that are described as reactive, see for example [187], where reactive planning is called feedback planning, but very often the sensor-actuator-coupling is neglected or intense data accumulation is required. In what follows, we will focus more on vision-based obstacle avoidance, since promising reactive obstacle avoidance approaches have been developed in the last 25 years using cameras.

2.3.1.1 Vision-Based Obstacle Avoidance

Cameras, in monocular or stereo configuration, are a popular sensor for obstacle avoidance for autonomous vehicles and mobile robots by providing a rich variety of information. A detailed discussion about the pros and cons of cameras is made in Chapter 3.1.1. The main information provided by the sensors used in the approaches above is metric depth, which is in contrast to many biological systems that rely on different sources of depth cues [60]. A good overview of potential depth cues used by biological systems is given in [225], where also a main distinction is made into the categories:

- Motion Parallax: the angle to closer objects changes faster than for distant objects when the camera is moved.
- Monocular Cues: like for example perspectives (vanishing point), relative sizes, known object sizes, occlusions.
- Stereo Vision: the depth is determined via triangulation.

Since stereo cameras, if calibrated, inherently provide depth information, they are often used very similarly to range sensors like RADAR or LIDAR, and data is accumulated to

generate maps. For example: a potential field approach with terrain classification is applied for a mobile robot in an outdoor scenario in [147], while the occupancy grid map is generated from stereo vision.
Before focusing on the category of motion parallax (mainly optical flow), an overview over approaches based on monocular cues is given.

Monocular Cues: Different monocular obstacle detection approaches have been proposed for mobile robots in the last three decades. In [195], the motor commands (turn-angle and forward speed) for a mobile robot are directly derived from object boundaries that are extracted from camera images using an edge detector and RGB/HSV filter. Horizontal boundaries indicate the relative distance and the direction of the motion is derived by turning toward the most distant border. In a similar manner, a pixel-wise classification of ground and obstacles is carried out in [290] based only on color. A monocular camera is also used in [73] to identify obstacles first by its shape and then calculate the angles to the object boundaries. Considering those, a new navigation target point is determined, which leads to a new turning angle. The work of [192] describes a different approach, where ground plane features are identified by using the homography principle (explained in Chapter 3.1.4). After that the detected ground plane is projected into an occupancy grid map via an inverse perspective transformation. In those ground-based cases, the depth cue is derived from the respective image row of the pixels.
An approach based on relative sizes is presented in [225], where a monocular camera on a UAV is used to observe the expansion of objects to detect and avoid collisions. Here, SURF features are used to extract the expanding key points in combination with template matching to determine the ratio of expansion. Therefore, no depth information is required, which cannot be provided by the monocular camera. This approach has a strong reactive character, since the sensor properties are taking into consideration, data is not accumulated and the motion response is directly derived from the detected expansion rate, which is strongly related to the time-to-collision that is explained in the following.

Time-to-Contact/Collision: A very important principle regarding monocular obstacle avoidance is the calculation of the Time-to-Contact/Collision (TTC). Basically, the TTC denotes: how many frames does it take until an object contacts the camera plane.
A comparison of different approaches for measuring the time-to-collision is provided in [18]. The huge advantage of the TTC calculation is that a monocular camera is sufficient and no calibration or no depth estimation is necessary. Lee claimed already in 1976 [188] that a human driver could benefit from a TTC information that tells him when to brake to avoid collisions. Closely related to the TTC is the optical flow, that belongs to the category of motion parallax depth cues.

2.3.1.2 Obstacle Avoidance based on Optical Flow

The shift of an image point between two consecutive images is called optical flow, while the first algorithms for computing it were already developed in the early 1980s [162, 199]. Using this principle, motions could be sensed with a monocular camera. Two types of optical flows can be determined:

1. **Dense optical flow**: the flow vectors are evenly distributed over the entire image like in [309].

2. **Sparse optical flow**: significant feature points are detected and matched between consecutive images.

It must be mentioned that dense optical flow is computationally expensive [234] and is still a challenging problem for moving cameras [227]. A comparison and performance evaluation of the early optical flow techniques can be found in [28]. Since then a number of improvements have been introduced to tackle problems like occlusions [21], respecting motion boundaries [250], and the preservation of discontinuities [96, 156].Warping strategies are introduced in [57] to increase the accuracy. Moreover, the fusion of the optical flow from a standard camera with the flow from an infrared camera is suggested in [231] for an increased robustness in urban areas.
The feature-based techniques, which provide sparse optical flow, need less computational time then the dense approaches and have no aperture problem [70]. Sparse optical flow is discussed in more detail in the context with image features in Chapter 3.1.3.
Already in 1989, Nelson proposed a method that makes use of the flow field divergence in order to avoid obstacles [229]. Despite the very limited computational power the approach could be verified with basic obstacle detection tests. Since then, the optical flow has been the basis for different reactive obstacle avoidance approaches.
Some of them use the TTC for obstacle detection like, e.g., in [65], where regions with similar flow are clustered and the TTC is estimated to identify dangerous regions for a collision-free flight of an UAV.

Approaches inspired from Biology: Early real time approaches used the time to contact to perceive obstacles along the central motion direction, while balancing peripheral flows to follow a corridor. This principle was inspired by biology, as insects are a rich source of inspiration for vision-based collision avoidance. They manage complex tasks based on visual motion information [270] despite being equipped with simply evolved eyes and brains and lagging stereo vision. Fruit flies detect emerging obstacles by their visual expansion with different motion strategies being elicited by the kind of expansion [276]. Honey bees fly collision free through narrow passages by balancing image speeds on two sides and even use image motion to estimate covered distances of several hundred meters [271]. Ground-based robots using this principle were build in the early 90s [256] and a mobile robot in [81] was able to move in a static environment collision free for up to 20 minutes with an average speed of 30 [cm/s]. Simple strategies were pursued due to very limited computational power. For example, a few approaches use only the full resolution at the center region and lower at peripheral regions [27, 283]. Therefore, the amount of data is decreased without the loss of necessary information.
Nowadays, insect strategies are used especially in small flying vehicles that even have sizes comparable to insects or birds [241], and therefore they are not able to carry fast and heavy computing devices. One of those micro air vehicle (MAV) achieves good results by avoiding regions with high optical flow [144].

Structure-from-motion: Methods similar to structure from motion (an introduction can be found e.g. in [201]) are utilized to overcome the missing depth information. One example is described in [268], where the motion direction is determined based on a depth histogram. The depth measurements are derived from optical flow in combination with the wheel speeds, which provide the metric scale. If the flow is not caused exclusively by the ego-motion of the camera, only the TTC is determined, which is the case for a dynamic obstacle.
A similar approach is described in [197], where a mobile robot creates an angular depth map from optical flow and wheel speeds. This representation is then compared to maps from sonar and laser data with the result that the depths from motion algorithm performs only slightly worse than the LIDAR method. More sophisticated approaches are introduced, e.g., in [255] for a UAV equipped with an IMU and a monocular camera, or in [149] for a self-driving car using a fish-eye camera and wheel odometry.

General Shortcomings of Optical Flow: A few shortcomings have to be taken into consideration, when using optical flow for collision avoidance. Rotations between successive frames are a typical problem [81] and have to be compensated. An often chosen possible solution is the use of an additional sensor like an IMU [78, 165], but the detection of rotations can also be handled visually with, e.g., the Zinf ego-motion estimation from [207]. Frontal objects are also problematic, as the main motion component is along the optical axis of the camera and therefore, the observable motion is small. Approaches based on monocular cues like [225] explicitly tackle this problem.
A further closely related shortcoming of the optical flow is the requirement of motion in general. The observed motion must have a certain magnitude to provide a useful signal to noise ratio. Combining optical flow with stereo vision [159] could solve this problem. Those approaches complement each other for autonomous vehicles, since the range of stereo vision is limited by the base distance between the cameras, but perform well in close ranges, whereas the optical flow benefits from large velocities, where the detection range of obstacles must be larger. Very good results were achieved in this way, e.g., in [245].
Dynamic obstacles denote a special case for optical flow-based approaches, as the relative motion between object and camera is observed. Except the TTC, all approaches mentioned until now require adaptations, when dynamic obstacles are present.

2.3.1.3 Dynamic Obstacles

All the previously named methods, using cameras or not, lack the ability to cope explicitly with dynamic obstacles. Therefore, different extensions were introduced like, e.g., in [135], where the potential field method does not only consider the relative position of an obstacle to create the repelling force but also the relative velocities. This additional information weakens further the direct sensor-actuator-coupling, since a sensor is required that concurrently measures distances to and velocities of objects and would provide metric sizes in the best case. In [122] inevitable collision states are introduced that are used in motion planning or for navigation to describe the dynamics of both the robotic system

and the obstacles. This requires an extensive knowledge about the environment and its dynamics. The same holds for methods that are settled in the velocity space [118, 232] of the system to take the actuator dynamics into account, whereas the sensor type and properties are completely neglected. The concept of Probabilistic Velocity Obstacles, which considers circular objects and their velocities together with uncertainties [179], are used in combination with a dynamic occupancy grid [83] for, according to the authors, “a reactive algorithm to perform obstacle avoidance” [132]. The dynamic occupancy grid, which is the input for the algorithm, is generated from LIDAR data and uses Bayesian filter techniques to establish a prediction functionality. Speaking of reactive algorithms, at least the problem of data accumulation arises here. Another approach with prediction ability using an occupancy grid map is described in [262]. It is stated as reactive, since a dynamic window approach adaptation is used to avoid local obstacles. Although the actuator properties are well integrated in this representation, it neglects the sensor and assumes that the velocity vector and motion heading for each moving cell is known.
A popular approach to identify dynamic objects is the fusion of depth information from stereo-vision with motion information from optical flow [126]. The work of [29] uses additionally a vehicle dynamics model to identify the state of moving vehicles and predict their driving paths. Nevertheless, it also cannot be considered as reactive approach and is mentioned here as typical example how dynamic objects are handled in the state of the art. A detailed overview of vehicle detection techniques can be found in [226].
It is difficult to find an approach in literature that is able to avoid collisions with several dynamic objects concurrently and that is designed, at least to a large extend, in accordance with the described properties of a reactive instinctive behavior.

2.3.2 Vision-Based Control and Navigation

Navigation denotes a robot’s task to come (collision-free) from one configuration/state to another one [77]. In the case of mobile robots or autonomous vehicles the question refines to: how should the robot be moved such that a goal configuration is reached.
Regarding the horizontal scheme of Figure 1.3, the navigation task is settled above the obstacle avoidance. Furthermore a distinction is made between local and global goals [86]. Recalling the initial part of this section, local goals are mainly interesting for reactive instinctive algorithms. A global goal or plan can consist of a sequence of local goal states. In comparison to obstacle avoidance additional information must be available to be capable of two sub-tasks:

1. The minimum requirement: a system is aware of being at the goal state, when it reaches it.
2. The current state and the goal state should be located in the used representation.

To fulfill the first requirement, a description of the goal state, which could be compared to the current state, must be available. This could be realized via feature points [68] or via GPS coordinates, like in the DARPA challenges [282]. For the second requirement, the system must either be able to localize itself in a global representation or map [281], or it can localize itself in relation to the goal state [45]. Regarding the characteristics of reactive approaches, the latter case is more suitable and very similar to navigation tasks that can be performed by a human driver without reasoning. Therefore, this kind of navigation is considered as instinctive/intuitive and can be seen as a relative positioning task:

- Relative to an **abstract property** like, e.g., lane-keeping. Even though only the lateral control is a positioning task.

- Relative to a **target pose** like, e.g., parking in a well-known environment (the garage at home).

- Relative to a **moving target pose** like, e.g., following a preceding vehicle, where in addition to the lane-keeping case also the longitudinal control (keeping a safe distance) is a relative positioning task.

The navigation principle where a robot positions itself in relation to an abstract property is used, e.g., in wall following approaches [298]. Due to simplicity this class of reactive navigation algorithms is not further considered in this work, while the focus is laid on the latter two.

2.3.2.1 Reactive Navigation with Fuzzy Logic

Fuzzy or neuro-fuzzy techniques are designed for problems where the non-linear dynamics or properties are not known or can only be modeled with high efforts, but can be described by a set of heuristic rules. An approach for autonomous parking is proposed in [87] that uses fuzzy logic to cope with uncertain and dynamic environments. Moreover, the algorithm tries to take the dynamics and limitations of the car-like robot into account, as well as uncertainties and capabilities of the sensor. In [269] a heuristic fuzzy-neuro network is developed to map ultrasonic data to robot velocities for navigating in an unknown environment. Furthermore, the work of [307] proposes a fuzzy logic controller with local target switching, which helps to avoid limit cycle paths, for reactive navigation of a mobile robot. At first sight, those approaches based on methods from computational intelligence are designed in accordance with the postulations from Section 1.1. The fuzzy controller can be considered as “an efficient representation of task-relevant information” and also data accumulation is avoided. However, it is debatable if the extensive use of heuristic rules coincides with “the use of only limited pre-knowledge or assumptions” or if the sensor and actuator space are really covered by the representation. Furthermore, the fuzzification decouples the sensor from the actuator - not in a way that the action cannot be directly derived from the sensor input, but the actual physical properties or measures are disregarded. The abstraction level is too high and the tuning effort for the rules limits its applicability.
Vision-based approaches are more promising since they are not heuristics-driven and interesting reactive navigation algorithms can be found in literature, as can be seen in the next subsection.

2.3.2.2 Visual Servoing

The underlying principle of vision-based control is the use of visual information in the feedback loop to complete positioning or navigation tasks [76].
For over 40 years [164] vision-based control techniques have been a crossover research topic between image processing and control theory. Hence, the characteristics of the visual sensor and of the robot have to be taken concurrently into consideration when designing the vision based control scheme. This fact makes those approaches very convenient for designing algorithms for reactive instinctive behavior.
Detailed survey papers about vision-based control or also called visual servo control or visual servoing are, e.g., [164], [68], and [69]. Basically, a distinction is made between two basic classes of visual servoing schemes and a hybrid one:

- 3D / position-based visual servoing
- 2D / image-based visual servoing
- 2.5D visual servoing

The first category, the position-based visual servoing (PBVS) or 3D visual servoing, formulates the control task in Cartesian coordinates, like [277]. Further subcategories are model-based and model-free 3D visual servoing [208] with the distinction that model-free techniques have no 3D information of the target [30].
The second category, the image-based visual servoing (IBVS) or 2D visual servoing, is the most interesting class of those three for reactive algorithms. The control task is settled directly in the image space with the advantage of an increased robustness against calibration errors [113] and requiring no target model. Therefore, the control law is designed in dependence of image properties like for example 2D feature points. However, also other image features or properties like straight lines [114], eigenimages [95], or image moments [67, 275] have been utilized.
Additionally, combinations of image-based and position-based visual servoing (2.5D visual servoing) were proposed, e.g., in [211]. The task function contains variables in the Cartesian space as well as in the image space to overcome shortcomings, like the possibility that in IBVS approaches features might leave the field of view of the camera, and use strengths of both visual servoing types, like the robustness of IBVS against calibration errors [210]. Furthermore, approaches exist, where the system switches between a IBVS and a PBVS controller [134].
A great variety of extensions has been developed for all visual servo classes regarding online parameter estimation, decoupling translations from rotations [82, 274], or optimal control [84, 19].
For 3D visual servoing, the epipolar geometry, see [201], is exploited like in [30] where the current pose is estimated by the essential matrix. If the target is planar or the motion between two images is a pure rotation, the homography principle, see Section 3.1.4, has to be utilized like in [116].
In the case of IBVS, the depth distribution can be estimated by structure from motion or by exploiting the epipolar geometry (if a 3D target model is available) [69]. Moreover, parameters of the interaction matrix, which is also called feature Jacobian as it maps the

robot's velocities into the image space, are approximated [209] or estimated numerically [186], by Broyden's method [167] or by neural networks [304].
Another interesting but exploratory work should be mentioned for completeness: kernel-based visual servoing tries to combine the usually separated tracking and control [170].

IBVS for Mobile Robots: Among a great variety of applications for vision-based control ranging from industrial to medical robotics [182], the most interesting field of application for this work is vision-based navigation for ground-based mobile robots or autonomous vehicles.
Since especially the IBVS class coincides with the requirements to a reactive instinctive behavior, the main task now is to derive motion commands from an up-to-scale representation, as no metric measurements are possible with a monocular camera without any assumptions or further knowledge, see Section 3.1.1 for more information. Different (more or less elegant) methods and strategies can be found in literature. However, it is not possible to cover all proposed methods and results, so that only a few (recent) approaches are presented in the following to give an idea of the diversity.
A reactive indoor navigation approach for a ground-based mobile robot using only a monocular camera is presented in [64]. Natural landmarks are first used to teach the mobile robot a path, so that it can then re-drive the taught route autonomously using motion commands directly derived from the image features via a Jacobian. The problem of the missing depth is overcome by reformulating the problem so that only the metric heights of the feature points above the ground have to be estimated during the training phase.
The work of [249] also utilizes an image memory to navigate, but in contrast to the previous work the goal is not to follow an exact path. A sequence of views describes a series of areas (with certain boundaries) the robot has to visit during its way to the goal position. Since the control objective is not an exact positioning, the authors denote this approach as a "qualitative visual servoing". Therefore, an exact determination of the scale is avoided, respectively, not necessary during the qualitative navigation.
In [104] a visual-path-following framework is evaluated in different urban outdoor environments. The path is again represented by a series of target images that are tracked by a monocular camera. Tests show that the scheme can successfully navigate also in environments with changing lighting conditions and is not disturbed by moving objects. However, the mainly reactive scheme uses a feature prediction technique based on local 3D geometry estimation to deal with tracking errors. The work in [20] presents a model predictive trajectory tracking approach for mobile robots that formulates the task as non-linear optimization problem in the image plane. This enables the explicit consideration of constraints. Moreover, the system model that is used for the prediction consists of a model for the camera and a model for the robot and therefore the sensor-actuator coupling is fulfilled. However, a huge drawback is the necessity to measure or approximate the depth of the feature points, which is not directly possible with the sensory equipment.
The problem of measuring the distance to feature points does not exist for appearance-based approaches. The work of [88] tries to utilize the information of an image (as defined by Shannon) directly in a visual servoing scheme. Therefore, an optimization

task is designed to maximize the mutual information between the current image and the goal image. This further results in an increased robustness against occlusions and illumination changes. The application of this mutual information visual servoing approach to path following is described in [89]. Real-world tests are described, where a non-holonomic vehicle follows a path that is defined by a sequence of images. However, only the angular control, for which an optimizer determines the steering angle by maximizing the mutual information, can be established by this approach.
A comparison of an IBVS and a PBVS scheme for path tracking and path reaching for a mobile robot is made in [75]. The IBVS controller performs better when the calibration is incorrect and is generally more accurate, but it does not work in all test applications. Large initial error are problematic due to the missing curvature feedback in the control law. The paper suggests that the pose-based controller is used for path reaching and the image-based for path tracking.

Vision based control techniques have a lot of potential to be used for designing reactive instinctive algorithms, as the relative position to the target state is measured directly without building an environment map by data accumulation. Unfortunately, the control law of the standard IBVS approach requires the depth values of the features, which impairs the direct use of the sensor. An elegant solution to insert scale information to the control scheme without constantly measuring a metric size is the use of epipolar geometry exploiting techniques like the homography principle.

2.3.2.3 Homography-based Techniques

One solution for a vision-based control scheme without estimating or measuring metric sizes is the use of a homography matrix [38], which is explained in detail in Chapter 3.1.4. The only requirements are that the target view must be known and the feature points have to lie on a planar surface [201]. Often the homography matrix is decomposed to determine the relative rotation and translation from the current camera position to the target position in Cartesian coordinates. This principle is used in [72] for a mobile robot that autonomously tracks a path described by a sequence of images.
One shortcoming of the decomposition is that it results in two physically feasible solutions from which one has to be chosen. A further drawback is that the transformation to Cartesian space impairs a direct use of the image data, which would be in accordance with properties of a reactive scheme. Moreover, a Cartesian representation requires additional processing with error prone parameters and increases the computational costs. An approach to avoid the decomposition is suggested in [194], where the motion of a unicycle robot is decomposed to three motion primitives. A controller is designed to execute those motion primitives by utilizing the entries of the homography matrix. However, the decomposition still has to be carried out for certain motions. A similar approach is described in [191], where two steps are used to first drive the mobile robot to the goal position and then to correct the orientation. Although the authors claim to be able to deal with uncalibrated cameras, they do not support it with experimental evidence.
A different concept is described in [38] which enables a decomposition-free visual servoing control for a 6-DOF robot by abiding by restrictions regarding the relative position of the goal pose to the plane of the image features. The same principle is utilized in [163] for a

relative positioning task of an underwater vehicle.
A navigation task - in the sense of a relative positioning - is definitely suitable to be executed via a reactive instinctive behavior. Vision-based control algorithm are a very promising concept for this, but the challenge then is to find an elegant and robust way to insert a metric size into the system for deriving an absolute motion command. Moreover, a suitable goal state description has to be designed that describes the target pose uniquely and that can easily be set in relation to the current pose.

2.3.3 Vision-Based Platooning

A special category of robot navigation tasks is characterized by a dynamic target pose. An example can be found in [135], where the potential field approach is extended to deal explicitly with a moving goal position.
A practical application of this navigation category is vehicle following or also called platooning - surveys can be found in [173, 43]. Another example is the VisLab Intercontinental Autonomous Challenge from 2010 [54], where a convoy of two vehicles drove mostly autonomously from Parma to the Shanghai World Expo 2010. The leading vehicle was controlled autonomously only on parts of the 16000 kilometer long route, whereas the following vehicle operated autonomously along the whole route by tracking the leader's path [53]. The path tracking was established by either GPS way-points, or vision-based pose estimation supported by LIDAR.
In [39] vision-based car platooning is realized by tracking the back of a leading vehicle. A homography matrix is calculated by comparing the current view of the tracked pattern to a target image. After that the homography matrix is decomposed to determine the relative translation and rotation, which is then the input for a path planner. Therefore, the control task itself is described by a Cartesian reference frame spanned by the preceding vehicle. Another example can be found in [92], where a homography decomposition is utilized for estimating the leading vehicle's position and also the velocity.

2.3.4 Concurrent Reactive Obstacle Avoidance and Navigation

Map-based motion planning for mobile robots combines both obstacle avoidance and navigation. Although a Cartesian map is often used as the preferred representation, on which the planner works, few reactive approaches exist that can tackle concurrently obstacle avoidance and navigation. One possibility is to extend a reactive obstacle avoidance algorithms like [299] by an additional constraint that for instance prefers one evasion direction. The popular potential field approach can also be extended to consider a target pose explicitly [135]. Another classic example is the vector field histogramm [46] that uses a polar histogramm representation, which denotes the distances to the surrounding objects together with the angular direction. In [272] the motion of a wheeled robot is planned in the velocity space using the dynamic window approach [121], while the distances to the target and to obstacles are measured with a laser scanner. A further approach for a unicycle robot is introduced in [185], where an odometry-based path tracking task is performed while Deformable Virtual Zones are created for reactive obstacle avoidance.

The static obstacles are detected by infrared proximity sensors so that both tasks could be combined in a Cartesian representation.
More challenging are environments that contain also moving obstacles. Already in [123] from 1994, a state-time graph is proposed as representation for a car-like robot moving in an environment with dynamic obstacles. However, neither a method nor a sensor are proposed from which this representation can be generated. Deriving a combined obstacle avoidance and navigation approach that is reactive and considers explicitly dynamic obstacles is very difficult, as the navigation requires a certain planning horizon, for which the behavior of the dynamic obstacles has to be predicted. Usually, a motion model for the objects is generated from observations over several time-steps, which impairs the reactive principles. When dealing with dynamic objects, it is at first crucial to identify them as such and then to check whether a collision is about to happen, but for planning it is also necessary to determine the speed of dynamic objects to react in an appropriate way or to consider them within the motion planning. A good survey can be found in [267], which provides an overview of sensors and approaches for object detection, velocity estimation and vehicle tracking.
An often used approach to handle dynamic objects is to combine stereo-vision with optical flow, like in the 6D Vision approach of Daimler [245]. The combination of image motions with 3D data is also often utilized for model-based tracking [91, 178] or to build map representations like occupancy grids [234]. Those representations could be then used to track moving objects, e.g. by particle filters [90], while the velocity is derived from the particles. However, those approaches lack the reactive character. It is hard to find reactive velocity estimation methods for dynamic objects, but different visual ego-motion estimation techniques meet the reactive requirements. Here, a monocular camera is, e.g., used to estimate the ego-rotation by observing distant image points [205]. However, the ego-velocity can only be determined up-to-scale. Hence, different approaches exist that determine the absolute ego velocity from a combination of stereo cameras with either sparse [238] or dense optical flow [161]. The theory of binocular image flows was already examined theoretically in 1986 by Waxman [301]. Methods based on this principle have the drawback that they are computationally expensive and rely heavily on the disparity determination. One of them is described in [218], while only object tracking results are presented but no estimates of the object velocities.

2.4 Contributions

After having reviewed the state of the art, the scope and goals for this thesis can be identified. Recalling the basic ‘Perception-to-Action’ principle (see Figure 1.1), the three core elements that have to be considered when implementing reactive instinctive behaviors are the sensor, planning and actuation part. This thesis aims neither to develop a new actuator/vehicle nor to conceptualize a new type of sensor, but evaluates existing ones.
None of the reviewed autonomous vehicles of Section 2.1 fulfills the requirements postulated in Section 1.2, but the ROboMObil that is introduced in Chapter 4 does. For this reason, the ROboMObil is the ideal test-platform for the validation of reactive instinctive approaches in Chapter 5.

From the wide range of sensor types listed in Section 2.2, cameras are identified as the most suitable sensor regarding the postulations of Section 1.3. However, the variety of camera configurations is large, so that the influence of different camera parameters is discussed and suitable configurations for reactive instinctive approaches are proposed in this thesis.
Furthermore, the planning part is identified as the most important element when implementing a reactive instinctive behavior. However, the obstacle evasion and planning algorithms reviewed in Section 2.3 coincide only partially with the postulations of Section 1.1 and very often lack the capability to explicitly consider dynamic elements. Therefore, this thesis develops efficient representations that comply with the postulations, e.g. a minimal data accumulation or a strong sensor-actuator coupling, in order to gain the advantages of reactive instinctive approaches, such as an increased robustness. Suitable tasks that can be fulfilled by reactive instinctive approaches are listed in Chapter 1.4 and, thus, a representation for each is proposed and evaluated:

- **Obstacle evasion**: an approach is developed that is capable of avoiding multiple dynamic obstacles concurrently without depth reconstruction to preserve a reactive character.
- **Relative positioning**: a reactive instinctive approach is presented that controls the relative position to a static or dynamic target pose (compare platooning), while requiring only a monocular camera.
- **Navigation**: a suitable representation for concurrent obstacle avoidance and navigation is proposed together with a reactive motion planner. This representation is not based on a Cartesian map and considers explicitly dynamic targets, while obeying the postulations for a reactive instinctive approach.

The realization of those tasks requires additional functionalities that lead to additional contributions of this thesis. First, a principle is introduced that enables the estimation of the Time-To-Collision for a single image point. Second, a method is proposed to estimate the relative pose to and the velocity of target object from a monocular camera. Third, a velocity estimation approach is developed that provides absolute velocity values for dynamic objects without relying on depth reconstruction.

http://www.springer.com/978-3-658-19086-6

Robust Perception from Optical Sensors for Reactive
Behaviors in Autonomous Robotic Vehicles
Schaub, A.
2018, XV, 267 p. 127 illus., 123 illus. in color., Softcover
ISBN: 978-3-658-19086-6