Audi
Dissertationsreihe

Efficient Resource Allocation For Automotive Active Vision Systems

Stephan Matzka

HOCHSCHULE
FÜR ANGEWANDTE
WISSENSCHAFTEN FH
INGOLSTADT

Stephan Matzka

Efficient Resource Allocation for Automotive Active Vision Systems

Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

1. Aufl. - Göttingen: Cuvillier, 2010
Zugl.: Edinburgh (Heriot-Watt University), Diss., 2009

978-3-86955-250-7

Audi-Dissertaionsreihe, Band 24

Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

1. Auflage, 2010
Gedruckt auf säurefreiem Papier

978-3-86955-250-7

Abstract

Individual mobility on roads has a noticeable impact upon peoples' lives, including traffic accidents resulting in severe, or even lethal injuries. Therefore the main goal when operating a vehicle is to safely participate in road-traffic while minimising the adverse effects on our environment. This goal is pursued by road safety measures ranging from safety-oriented road design to driver assistance systems. The latter require exteroceptive sensors to acquire information about the vehicle's current environment.

In this thesis an efficient resource allocation for automotive vision systems is proposed. The notion of allocating resources implies the presence of processes that observe the whole environment and that are able to efficiently direct attentive processes. Directing attention constitutes a decision making process dependent upon the environment it operates in, the goal it pursues, and the sensor resources and computational resources it allocates. The sensor resources considered in this thesis are a subset of the multi-modal sensor system on a test vehicle provided by Audi AG, which is also used to evaluate our proposed resource allocation system.

This thesis presents an original contribution in three respects. First, a system architecture designed to efficiently allocate both high-resolution sensor resources and computational expensive processes based upon low-resolution sensor data is proposed. Second, a novel method to estimate 3-D range motion, efficient scan-patterns for spin image based classifiers, and an evaluation of track-to-track fusion algorithms present contributions in the field of data processing methods. Third, a Pareto efficient multi-objective resource allocation method is formalised, implemented, and evaluated using road traffic test sequences.

Für meine Eltern Josefine und Leopold Matzka.

Acknowledgements

In the course of writing this thesis I have enjoyed the support and encouragement of a number of great people whom I would like to thank here.

First and foremost I want to thank my supervisors Professor Dr. Andrew M Wallace and Professor Dr. Yvan R Petillot. Their constant support, insights, and valuable thoughts contributed greatly to this thesis. I truly cannot imagine a better team of supervisors.

I would like to express my thanks to the many friends and colleagues who shared their thoughts on my thesis, most notable Stefanie Angerer, Andreas Hermann, Dr. Zsolt Husz, Cüneyt Kaya, Stefan Lutz, and Dr. Carlos Spörhase. My brother Dr. Jürgen Matzka deserves my special thanks for his support throughout the entire process. Your advice was not only invaluable, but also taken.

The examiners of this thesis, Dr. Thierry Chateau and Dr. Daniel Clark deserve my sincere thanks for agreeing to examine my thesis and sharing their insights with constructive criticism, new thoughts, and encouragement.

Most of the research was carried out at the Ingolstadt Institute for Applied Research under the guidance of Professor Dr. Johann Schweiger, whose optimism and steady support has been the backbone of my work.

I want to thank the department for advanced driver assistance systems at Audi AG for funding the research and providing the test-vehicle. Of all people at Audi AG, I owe most to Dr. Richard Altendorfer, Dr. Björn Giesler, Norbert Keppeler, Paul Sprickmann Kerkerinck, Dr. Uwe Koser, and Professor Dr. Alfred Quenzler.

My heartfelt gratitude goes to my family, who has been a constant source of support and encouragement, offering a retreat when needed, but foremost a place to share the many joys of life. I am also indebted to Claudia Römer for forgiving me the many hours I spent with the writing of this thesis when instead I should have admired her beauty and enjoyed her sparkling wit.

Stephan Matzka

October, 8th 2009

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations

RT	road type
TP	traffic participant
ADTF	Automotive Data and Time Triggered Framework
C	classification
CGL	corpus geniculatum laterale
DGPS	differential global positioning system
DoG	Derivative of Gaussian
DS	diamond search
ECU	electronic control unit
EFK	extended Kalman filter
FMEA	failure mode and effects analysis
fps	frames per second
FS	full search
HSV	hue-saturation-value
HVS	human visual system
LR	lower right corner
LRR	Long range radar
MPEG	moving pictures expert group

MV	motion vector
PCA	principal component analysis
PCS	point cut search
PFC	prefrontal cortex
PMVFAST	predictive motion vector field adaptive search technique
PTZ	pan-tilt-zoom
RGB	red-green-blue
SAD	sum of absolute differences
SC	superior colliculus
SLAM	simultaneous localisation and mapping
SRR	Short range radar
TTC	time-to-collision
UL	upper left corner
US	Ultrasonic sensor
UTC	universal time, coordinated

List of Publications

Parts of this thesis are based on papers presented at scientific conferences. These papers are listed in the following:

- Stephan Matzka, Yvan R. Petillot, and Andrew M. Wallace. Fast motion estimation on range image sequences acquired with a 3-d camera. In Proceedings of the British Machine Vision Conference, volume II, pages 750–759. BMVA Press, 2007.
- Stephan Matzka, Yvan R. Petillot, and Andrew M. Wallace. Determining efficient scan-patterns for 3-d object recognition. In Proceedings of the 3rd International Symposium on Visual Computing, Lecture Notes in Computer Science 4842, pages 559–570. Springer, 2007.
- Andreas Hermann, Stephan Matzka, and Joerg Desel. Using a proactive sensor-system in the distributed environment model. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, pages 703–708, 2008.
- Stephan Matzka and Richard Altendorfer. A comparison of track-to-track fusion algorithms for automotive sensor fusion. In Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pages 189–194, 2008.
- Stephan Matzka, Yvan R. Petillot, and Andrew M. Wallace. Efficient resource allocation using a multiobjective utility optimisation method. In ECCV Workshop on Multi-Camera and Multi-modal Sensor Fusion, 2008.
- Stephan Matzka and Richard Altendorfer. Multisensor Fusion and Integration for Intelligent Systems, chapter A Comparison of Track-to-Track Fusion Algorithms for Automotive Sensor Fusion. Springer, pages 69–82, 2009.

Symbols

In this thesis a consistent set of symbols is used.

The coordinate symbols are described in Tab. 1

Symbol	Unit	Definition
x	m	longitudinal distance from the reference centre
y	m	lateral distance from the reference centre
z	m	elevation difference from the reference centre
i	px	horizontal pixel distance from upper left corner
j	px	vertical pixel distance from upper left corner
$\vec{p}$	m	position relative to the reference centre
$\vec{p}$	px	position relative to the top-left pixel

Table 1: Symbols for coordinates.

The data level quantities are defined in Tab. 2

Symbol	Unit	Definition
$l(i,j)$	cd/m^2	Luminance at pixel (i,j)
L	cd/m^2	Luminance matrix
$r(i,j)$	m	Radial distance at pixel (i,j)
R	m	Range matrix
$\vec{m}_{l(i,j)}$	px/s	Motion vector at pixel (i,j) in L
$\vec{m}_{r(i,j)}$	px/s & m/s	Motion vector at pixel (i,j) in R
M	px/s & m/s	Motion matrix

Table 2: Symbols for data level features.

Symbol	Definition
$\mathcal{TP}_n$	Traffic participant type
$\mathcal{RT}_m$	Road type
$P(\mathcal{TP}_n)$	(Prior) Probability for $\mathcal{TP}_n$
$P(\mathcal{TP}_n\|\mathcal{RT}_m)$	Probability for $\mathcal{TP}_n$ given road type $\mathcal{RT}_m$
$P(\mathcal{TP}_n\|C)$	Probability for $\mathcal{TP}_n$ after positive detection/classification
$P(\mathcal{TP}_n\|\neg C)$	Probability for $\mathcal{TP}_n$ after negative detection/classification
$P(C\|\mathcal{TP}_n)$	True positive rate of a classifier for $\mathcal{TP}_n$
$P(C\|\neg\mathcal{TP}_n)$	False positive rate of a classifier for $\mathcal{TP}_n$
$P(\neg C\|\mathcal{TP}_n)$	False negative rate of a classifier for $\mathcal{TP}_n$
$P(\neg C\|\neg\mathcal{TP}_n)$	True negative rate of a classifier for $\mathcal{TP}_n$
C	Positive detection/classification (true positive *and* false positive)
$\neg C$	Negative detection/classification (true negative *and* false negative)
$\mathcal{UC}$	Uncertainty
S	Saliency
s	Severity
η	Criticality

Table 3: Symbols for the reasoning process.

Symbol	Definition
$\mathcal{S}$	Sensor
$\mathcal{R}$	Region
$\mathcal{C}$	Classifier
$\mathcal{P}$	Priority
$\mathcal{A}_{\mathcal{S}_n}$	Partial allocation for sensor $\mathcal{S}_n$ (cf. Eq. 7.1)
$\mathcal{A}$	Allocation (cf. Eq. 7.3)
$\mathcal{A}^*$	Optimal allocation
Ω_n	Objective n
Ω	Objective set $\Omega = \{\Omega_1, \ldots, \Omega_{N_\Omega}\}$
$\mathcal{U}_n(\mathcal{R}_m)$	Utility for region $\mathcal{R}_m$ using objective Ω_n
$\mathcal{U}(\mathcal{R}_m)$	Combined utility for region $\mathcal{R}_m$ using objective set Ω
$\mathcal{U}(\mathcal{S}_n, \mathcal{R}_m)$	Combined utility for observing $\mathcal{R}_m$ using $\mathcal{S}_n$
$\mathcal{U}(\mathcal{A}_{\mathcal{S}_n})$	Combined utility for partial allocation $\mathcal{A}_{\mathcal{S}_n}$
$\mathcal{U}(\mathcal{A})$	Combined overall utility for allocation $\mathcal{A}$

Table 4: Symbols for the decision making process.

Chapter 1

Introduction

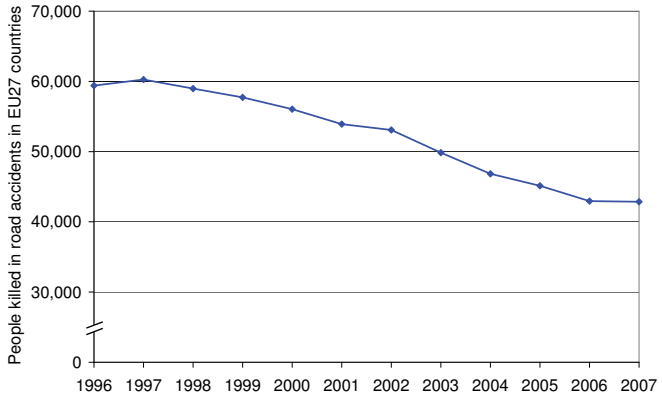
In this thesis an efficient resource allocation for automotive vision systems is proposed. Our system is capable of efficiently allocating both sensor resources and computational resources towards relevant regions in the environment. The notion of allocating resources implies the presence of processes that observe the whole environment and that are able to efficiently direct attentive processes. Directing attention constitutes a decision making process dependent upon the environment it operates in, the goal it pursues, and the sensor resources and computational resources it allocates.

In the following, a motivation for our investigations is given in section 1.1, with the initial situation pointed out in section 1.2. An overview of our system is given in section 1.3. The original contribution of our proposed system is presented in section 1.4. The introductory chapter closes in section 1.5 with an outline of the thesis.

1.1 Motivation

Every day hundreds of millions of people all over the world travel on roads. At the same time a great variety of traffic participant types exists, ranging from pedestrians to lorries. This great extent and diversity of individual mobility on roads cannot come about without a noticeable impact upon peoples' lives, including traffic accidents resulting in severe, or even lethal injuries.

Considering the road traffic accidents statistics for all 27 European Union countries (EU27, [1]) given in Fig. 1.1a, the number of lethal injuries in road traffic accidents shows a steady decline since 1997, but is still considerable with 42,854 people killed in road traffic

a) 

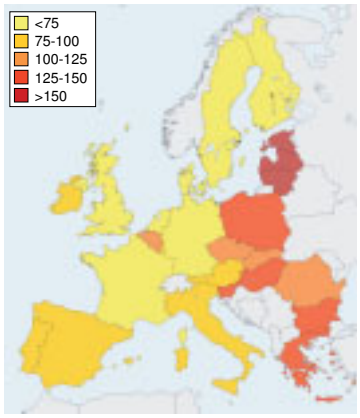
b) 

Figure 1.1: Road traffic accident statistics for EU27 countries. Figure a) gives the gradual decline of people killed in road traffic accidents in the last decade. Figure b) shows the number of people killed in road traffic accidents per million inhabitants for each country. Source: Eurostat [1].

accidents in 2007. Beyond this more than 9,000 people are injured in road traffic accidents on an average day. These substantial numbers result in a severe impact on the lives of EU citizens, as

> *"... one in 3 citizens will need hospital treatment during their lifetime and one in 80 citizens will end their life 40 years prematurely due to road crashes."* (European Commission [2])

Prompted by these findings, the Swedish parliament established the *Vision Zero* policy in 1997 [3], which is aimed at reducing the number of traffic deaths to zero by 2020. Other European countries, such as the United Kingdom and the Netherlands, followed with similar programmes, believing that the number of casualties can effectively be minimised instead of accepting severe accidents as an inevitable result of road traffic. This belief is also supported by the number of people killed in road accidents per million inhabitants shown in Fig. 1.1b). There, countries such as Sweden, the United Kingdom, and the Netherlands show a comparably low relative number of severe road traffic accidents.

Recognising that

> *"90 percent of road accidents are attributable to human error"* (Vliet and Schermers [4])

a key requirement of road safety programmes is to provide means to prevent people being killed or seriously injured in a road traffic accident, even if it is caused by human error [3].

Proposed measures to ensure this are roads designs that are self-explaining and impeding accidents, speed limitations, and passive safety systems, but also active driver assistance systems in cars.

Human error, besides deliberately risky driving, can generally be attributed to either perceptual errors, or a lack of attention (cf. Green *et al.* [5]). Perceptual errors usually occur if the visual contrast of critical objects and traffic participants is low due to dim light, glare, or adverse weather conditions such as rain or fog. Note however, that the same adverse visual conditions apply to optical sensors such as video cameras, and thus to vision-based driver assistance systems.

The second, more frequent, error is the lack of attention towards a critical object, or traffic participant. According to Green *et al.* [5] this lack of attention can originate from a number of impairing and distracting factors:

- internal impairments, e.g. fatigue, or drugs.
- internal distractors, e.g. strong emotions, or contemplation.
- external distractors, e.g. conversation, or surprising events.

It is apparent that the above impairments and distractors do not extend to computer systems. Therefore it appears desirable to assist a human driver with a computer vision system that is designed to continually detect and classify other traffic participants. This information is then provided to driver assistance systems.

This thesis investigates an automotive active vision system able to provide information about other traffic participants to the vehicle's environment model. Yet, in real-life scenes, the amount of sensor data gained by a standard video camera alone easily exceeds the computational performance of current embedded automotive hardware. This problem can be addressed by focusing computationally expensive tasks, such as object classification, to a fraction of the original sensory data. This process of directing visual attention is also observable for the human visual system, as only about 0.3% of the information carried through the optic nerves reaches attentive scrunity according to Anderson *et al.* [6].

However, attention implies the presence of non-attentive processes that observe the whole environment, and that are able to efficiently direct attentive processes. Directing attention constitutes a decision making process dependent upon the environment it operates in, the goal it pursues, and the sensor resources and computational resources it allocates. These dependencies are illustrated for our system in section 1.2 below.

1.2 Initial situation

In the following section, a short overview of the initial situation is given. First, the road traffic environment in which our ego-vehicle is situated is described in section 1.2.1. Second, the goal that we aim to achieve with our proposed system is defined in section 1.2.2. Third, the sensor configuration of our test-vehicle used for the design and evaluation of our active vision system is presented in section 1.2.3.

1.2.1 Environment

An environment can be described using an ontology as a form of knowledge representation (e.g. Uschold and Gruninger [7]). This concept has been transferred into information sciences, where an ontology has been defined as

"a formal representation of entities and their relationships within a domain." (Gruber [8, 9])

We have developed an ontology shown in Fig. 1.2 as a model for the road-traffic environment.

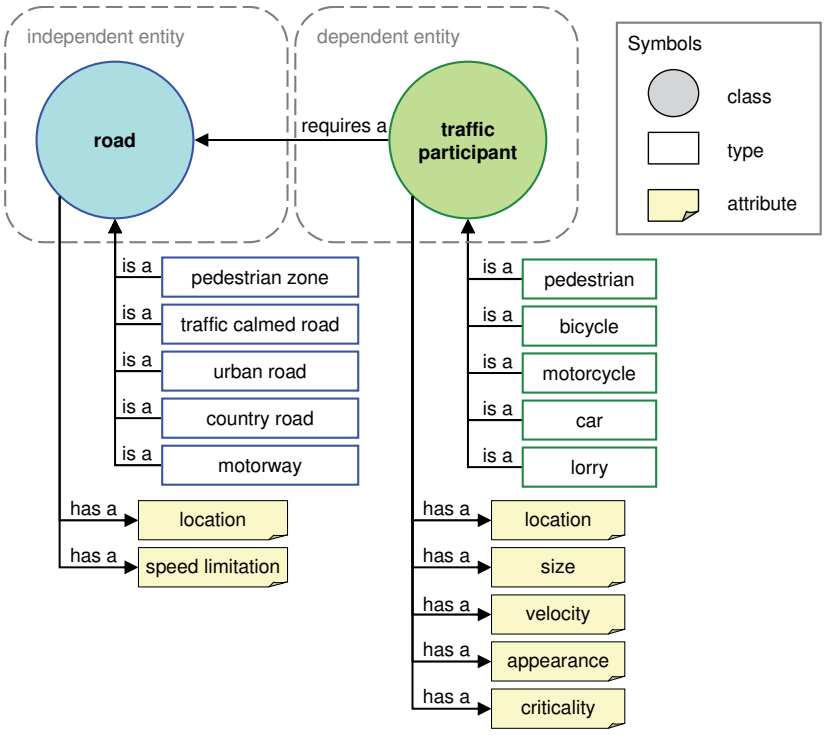


Figure 1.2: Ontology of road-traffic environment. Circles represent classes with a number of assigned types drawn as rectangles. Every instance of a class also has all attributes of that class.

It can be seen from our ontology in Fig. 1.2 that two classes exist: road and traffic participant. A road can exist independently, whereas a traffic participant inherently requires a road. The classes are subdivided into several road types and traffic participant types respectively. Every instance of a class is always assigned exactly one type and possesses all of the class's respective attributes.

The presented ontology provides a knowledge representation of the environment, that must be considered during system design, and is also used as a basic structure for our implementation. However it does not indicate how the attributes of instances are determined. Moreover it does not state what our goal of directing visual attention is.

1.2.2 Goal

Before the goal of directing attention can be stated, the goal of operating a car in a road-traffic environment must be defined. We consider this goal to be to

- safely participate in road-traffic while minimising the adverse effects on our environment.

This goal implies both the presence of an ego-vehicle and an environment. Following our ontology, the environment consists of a road and traffic-participants including our ego-vehicle. Beyond the ontological definition, our environment is also an ecosystem. In order to protect the environment while participating in road-traffic, two of premises can be postulated:

- In order to protect the passengers of the ego-vehicle and other traffic participants, do not collide with other traffic participants (collision avoidance).
- In order to minimise adverse effects on the environment, operate efficiently (environmentally friendly).

Notwithstanding the importance of both premises, our proposed system is designed for *collision avoidance* to ensure safety of both ego-vehicle and other traffic participants. However, safe operation of a vehicle also implies energy efficiency. For example, unnecessary braking is avoided by maintaining an adequate safety distance to other traffic participants, as are traffic jams caused by car accidents. Another frequently ignored factor is that the production of a car is an energy-intensive process. The manufacturing of a medium-sized car consumes approximately 74 GJ of energy and causes a CO_2 emission of 2.8 t, the

latter matching the CO_2 emission of driving a medium-sized car for 18,600 km [10]. This manufacturing process has to be repeated at least in part in case of a car accident.

In order to achieve collision avoidance, four subsequent phases are identified:

1. Information about other traffic participants is obtained.
2. Obtained information is used to build a model of the current environment.
3. A strategy for collision avoidance in the given environment is devised.
4. Actions are enforced based upon the collision avoidance strategy.

Our proposed resource allocation system is situated in the first stage of obtaining information about other traffic participants. Beyond this, our investigations also include the definition and implementation of a distributed environment model described in Hermann *et al.* [11].

The control exerted by the resource allocation is restricted to the set of sensor resources and classification modules. This is in contrast to systems that determine actual object avoidance strategies, or systems that can actively induce an emergency stop or an avoidance manoeuvre. The restriction towards collision avoidance and there towards observing the environment affects the statement of our goal. The latter can be rephrased to

- protect the passengers of the ego-vehicle and other traffic participants by reducing uncertainty about traffic participants with whom a collision is possible.

The rephrased goal statement implies that some regions in the environment are more relevant than others, depending upon whether the region contains other traffic participants. Moreover, not all traffic participants can be assumed to be equally relevant. First, traffic participants and objects with whom a collision is possible are more relevant than those with whom this is not the case. Second, traffic participants and objects which are dangerous for the ego-vehicle are more relevant for passenger safety. Third traffic participants which are vulnerable to the ego-vehicle are more relevant for traffic participant safety.

Apart from the relevance of the region itself, it is important to choose a region where observation reduces uncertainty about the candidate region. Observation implies the use of exteroceptive sensors, which are described in the following.

1.2.3 Sensor Configuration of the Test-Vehicle

The proposed system obtains its sensor-level data from a multi-modal sensor system mounted on a test-vehicle provided by Audi AG consisting of

- two high-resolution video cameras
  - one pan-tilt-zoom camera
  - one fixed camera
- one 3-D camera (photonic mixer device, PMD)
- one time-of-flight laser scanner
- two short-range radars (SRR)
- one long-range radar (LRR)
- eight ultrasonic sensors (US)
- one differential global positioning system (DGPS)

In Fig. 1.3a the test vehicle provided by Audi AG is shown, in Fig. 1.3b the maximum detection distances and aperture angles of the different sensors is illustrated.

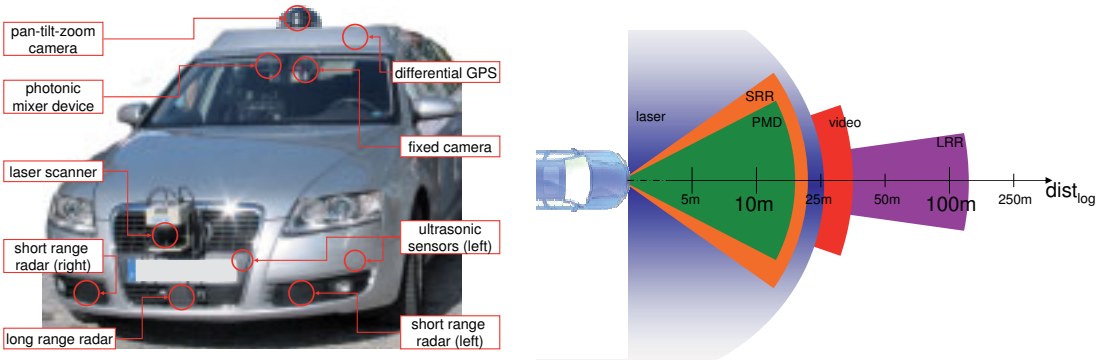


Figure 1.3: In a) the test-vehicle used to acquire road traffic scenes used for evaluation is shown. Figure b) the distance and aperture angles of the sensor array are drawn on a logarithmic scale.

Our selection of sensors from the test-vehicle's multi-modal sensor configuration is presented in section 1.3 below.

1.3 System Overview

In this thesis an effective resource allocation for an automotive active vision system is presented. A literature review shows that a variety of active vision systems exist. Despite the differences in the considered systems, all approaches encounter the same fundamental questions on system design:

- What is the overall system architecture?
- Which objectives are considered during optimisation?
- How are objectives determined for every candidate region?
- Which decision making concept is chosen to perform multi-objective optimisation?
- How is the complexity of the system reduced or, if this is not possible, handled?
- Is the system required and capable to fulfil real-time constraints?

Our active vision system processes data over various stages, beginning at sensor level and increasing both in level of abstraction and in significance towards allocation level (cf. Fig. 1.4). The system is organised using five levels of abstraction from sensor level towards allocation level:

1. Sensor level, containing raw sensor data representations.
2. Data level, containing the results of low-level sensor data processing.
3. Semantic level, containing semantic data resulting from high-level data interpretation
4. Reasoning level, containing combined semantic data to be used for. reasoning
5. Allocation level, containing the system's current resource allocation.

An increasing level of abstraction is highly desirable to maximise the system's efficiency, yet requires a set of serial processing steps. In order to mitigate the latency associated with serial processing, data processing tasks are run in parallel for every level of abstraction. Parallel processing requires the independence of the executed processes, which necessitates the use of individual data representation objects (drawn as parallelograms in Fig. 1.4) in every level of abstraction. Each representation object is updated by a single or multiple processes, providing data for subsequent processing steps.

The sensors used in our proposed system are a selection from the multi-modal sensor system mounted on a test-vehicle provided by Audi AG (cf. section 1.2.3). Our presented system performs computationally expensive tasks such as object detection and object

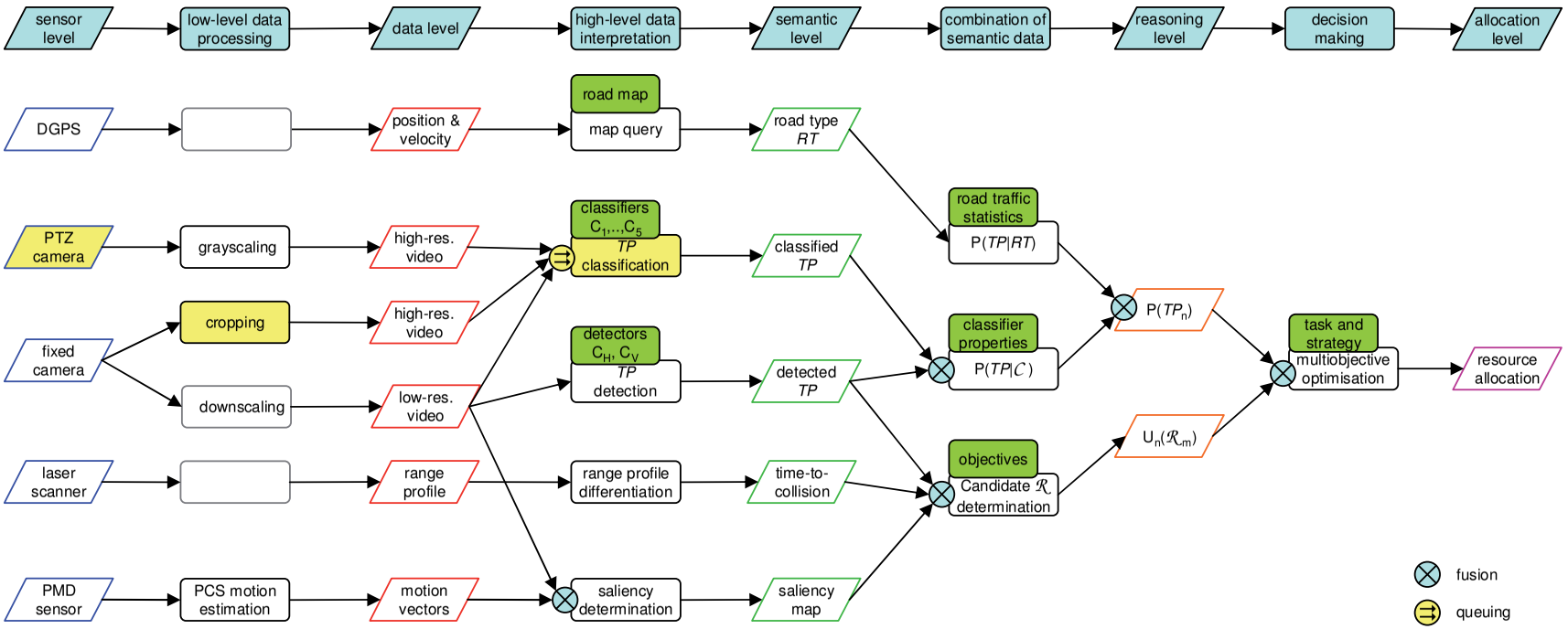


Figure 1.4: Block diagram of proposed system. Parallelograms indicate a data representation, whereas rounded boxes represent processing steps between two data representations (empty gray boxes indicate that no processing is necessary). Green containers indicate prior knowledge used in our proposed system. Circles represent pre-processing steps such as fusion (blue) and queuing (yellow). A yellow background colour indicates that the respective sensor or process is directly influenced by our resource allocation system, whereas all non-yellow sensors and processing steps work independently.

classification only on low-resolution data or on focused regions of high-resolution data. Controllable sensors acquire high-resolution data only for regions determined by the sensor-resource allocation concept, reducing the amount of sensor data in the system. This is in contrast to related systems discussed in section 2.4, where a high-resolution representation of the entire environment is required to determine regions of interest.

All processes inside the system are designed to fulfil soft real-time constraints, in the sense that all processes are designed to terminate within a given cycle time at which the proposed system operates. If a process exceeds the current cycle's deadline, the subsequent data representation object is not updated. Dependent subsequent processing steps will then pause until the update of the outdated data representation object is performed. This architecture also ensures that the most current data representation is made available to subsequent steps. Computationally inexpensive processes such as bottom-up saliency determination increase the system's reactivity to changes in the environment even if traffic participant detection or classification processes fail to terminate prior to the current cycle's deadline.

The proposed system is designed to avoid the problem of single points of failure. If any data representation object is outdated, the system is able to continue operation, albeit at the expense of decision making quality. Two system failures can be identified as most problematic, however. First, failure of traffic participant classification is critical because driver assistance systems are no longer provided with updated traffic participant positions and classes. Second, failure of the resource allocation system itself presents a problem, which is partly mitigated by a graceful degradation of both sensors and computational resources towards a static operation mode using a predefined resource allocation scheme.

The system design itself, two data processing methods, and the resource allocation process present original contributions and are discussed in section 1.4 below.

1.4 Contribution

The contribution of this thesis is divided into three aspects: the proposal of a novel system design for automotive vision system, extensions to sensor data processing algorithms, and the formalisation and evaluation of decision making in the resource allocation process.

System Design

Our proposed system is organised in five levels of abstraction, extending the four-layered architecture presented in Matzka *et al.* [12]. This architecture ensures that the amount of processed and transferred data decreases as the level of abstraction increases. The reduction of processed data lowers the computational demands on the vehicle's electronic control units and the reduction of transferred data reduces the load of the vehicle's bus system. In order to counter the latency caused by serial processing over multiple levels, processes within the same levels are run in parallel. In addition semantic information is made available to driver assistance systems in the third out of five levels, with both sensor level and data level processes designed to be computationally inexpensive.

Sensor Data Processing

We present two extensions to existing sensor data processing algorithms. First, an extension of the PMVFAST method to estimate 2-D motion vectors towards the PCS method is published in Matzka *et al.* [13] that efficiently estimates 3-D motion vectors in range maps is presented in section 4.5.2. Second, the use of a sparse input of single scanlines to be used in 3-D spin image object classification. The generation of suitable sparse scanlines is described and evaluated in Matzka *et al.* [14] and is presented in section 5.3. Beyond this, the fusion of correlated pre-filtered radar tracks is investigated in Matzka and Altendorfer [15, 16].

Formalisation and Evaluation of Resource Allocation

The central contribution of this thesis is the formalisation and evaluation of the decision making process required for resource allocation first presented in Matzka *et al.* [12], extending existing active vision systems discussed in section 2.4. Our proposed system is novel in the respect that it combines a formal, Pareto efficient decision making method with bottom-up and top-down information acquired using low-resolution data. This is in contrast to methods presented in the literature selecting regions of interest from high-resolution data. An optimum decision making strategy is determined and the problem of decision making complexity is solved by presenting efficient search heuristics to determine the allocation with the highest estimated utility.

1.5 Thesis Outline

This thesis is organised in chapters corresponding to the levels of abstraction of our proposed system as shown in Fig. 1.5.

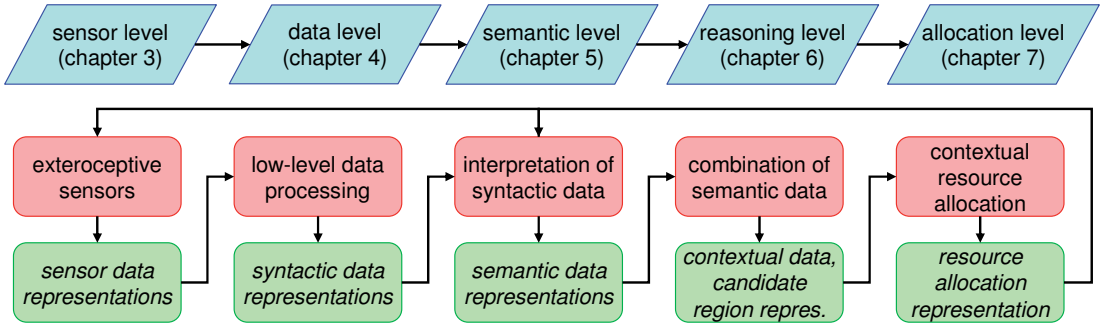


Figure 1.5: Organisation of the thesis corresponding to the levels of abstraction in the system overview given in Fig. 1.4. Red boxes show processing steps in the system, green boxes point out the resulting data representations.

After a review of integral parts and existing concepts for active vision systems in chapter 2, sensors and their data representations are covered in chapter 3. Low-level data processing steps towards a set of syntactical environment descriptions are given in chapter 4 on data level modules. Chapter 5 describes the bridging of the semantic gap, resulting in a set of relevant semantic data representations. In chapter 6 the combination of semantic data into a contextual data representation and our candidate region determination method is described and evaluated. The contextual resource allocation concept is described, evaluated, and discussed in chapter 7. Our conclusions and an outlook on future work are given in chapter 8.

Chapter 2

Literature Review

In this chapter fundamental methods as well as state-of-the-art systems in the field of active vision are discussed. Active vision is a computer vision paradigm stated to have four characteristics by Crowley [17]: continuous operation, filtering of information, operation in real-time, and control of processing.

For our given problem of an automotive active vision system, this can be stated to be a system able to allocate sensor resources and computational resources towards the regions with the highest attentional claims at the present moment. We believe that an efficient active vision system requires three integral parts to exhibit the above characteristics: controllable sensors, an object detection and classification system, and a decision making system to allocate sensor resources and computational resources.

This review focuses on these integral parts, as well as discussing existing active vision systems. For a comprehensive description of general methods in the field of computer vision, in which this thesis is located, the reader is referred to the works of Ballard and Brown [18], and more recent by Forsyth and Ponce [19].

The literature review is organised as follows: sensor systems used on automotive platforms are discussed in section 2.1, an overview of object detection and object classification methods for both 2-D data and 3-D data is given in section 2.2, and decision making methods are reviewed in section 2.3. Active vision systems presented in the literature are discussed in section 2.4.

2.1 Automotive Sensor Systems

Exteroceptive sensors constitute the first integral requirement for both autonomous vehicles and vehicles equipped with driver assistance systems. According to Mosby [20], exteroceptive sensors are responsive to stimuli that originate from outside. Sensor configurations for both autonomous vehicles and driver assistance systems are described in sections 2.1.1 and 2.1.2, followed by a discussion of the considered automotive sensor systems in section 2.1.3.

2.1.1 Autonomous Driving Systems

In the literature a wide range of autonomous driving systems is described. Early autonomous driving systems can be found in the field of car-like mobile robotics. Later autonomously driving vehicles have shown that a successful application of concepts from mobile robotics is feasible. Only recently, the development of autonomous vehicles has shown substantial progress, which can in part be attributed to the DARPA challenges described below. In the following the sensor systems of existing car-like mobile robots and autonomous vehicles are presented and discussed.

Car-like Mobile Robotics

In the field of car-like mobile robotics, exteroceptive sensors are required to navigate in known environments and to explore unknown environments. The classic problem associated with mobile robots is that of simultaneous localisation and map building (SLAM), which is required for both building consistent maps of the robot's environment and collisions avoidance, e.g. Dissanayake *et al.* [21], and Montemerlo *et al.* [22]. As a basis for localisation, mapping, and navigation applications, mobile robots are equipped with multi-modal sensor systems. In the literature, mobile robots' sensors are already described in the 1990s, e.g. Everett [23], and Borenstein *et al.* [24].

Sensor configurations naturally vary between robots depending on their individual task profiles. A frequent multi-modal sensor configuration of a car-like mobile robot described in the literature consists of an array of ultrasonic range sensors, a video camera, and a laser scanner which is also the standard sensor configuration of the Pioneer robots (cf. Fig. 2.1) largely used in robotics research.

The specification of the video camera is dependent on the task and thus can be either

colour or grayscale, can be a monocular or a stereo camera, and can either be fixed or controllable via a pan-tilt-zoom mechanism. The same applies for the laser scanner, which is mostly a single-beam laser scanner either mounted on a fixed platform (e.g. Montemerlo *et al.* [22]), a pitching platform (e.g. Frintrop *et al.* [25]), a rotating platform (e.g. Kohlhepp *et al.* [26]), or mounted sideways on a rotating platform (e.g. Brenneke *et al.* [27]).

Car-like mobile robots can also be equipped with a differential global positioning system in addition to relative odometry sensors to determine their absolute current position. In Fig. 2.1 car-like mobile robots used for early testing of automotive active vision systems are displayed. The robots are equipped with a PTZ video camera, sonars sensors, a laser scanner, odometry, and a DGPS.

a)  b) 

Figure 2.1: Our mobile robots used for early testing of automotive active vision systems equipped with a multi-modal sensor system. Figure a) shows the robots during initialisation, in b) the robot in the centre overtakes the robot in front autonomously.

Autonomous Driving Vehicles

Early autonomously driving vehicles to operate on public roads have been investigated in the *Prometheus* project in 1986. Later, research on autonomous cars has been invigorated by the Grand Challenges in 2004 and 2005, and the Urban Challenge in 2007 organised by the US Defense Advanced Research Projects Agency (DARPA). In the following, an overview of the sensor systems used in these vehicles is given.

Prometheus In the *Prometheus* project and its successor projects autonomous vehicles to drive on public roads have been investigated. The projects' results have been demonstrated in the test vehicles *VaMoRs-L* in 1986, and *VaMoRs-P* in 1994. While *VaMoRs-L* is a 5-ton van described by Dickmanns *et al.* [28], *VaMoRs-P* is a passenger car presented

in Dickmanns *et al.* [29], and Behringer and Müller [30]. Both vehicles rely on a two-camera sensor system: a wide-angle camera to detect the road and close obstacles, and a controllable focused camera to detect objects further away. The camera platform of both vehicles can be seen in Fig. 2.2.

a)  b) 

Figure 2.2: Camera system of the autonomous VaMoRs vehicles. Figure a) gives a schematic view of the VaMoRs-L van, Figure b) shows the camera platform of the VaMoRs-P vehicle during operation. Source: Behringer and Müller [30].

DARPA Urban Challenge 2007 The DARPA Urban Challenge is a research and development program for autonomous vehicles. In contrast to the desert courses used for the DARPA Grand Challenges in 2004 and 2005, the Urban Challenge 2007 featured an urban course situated in a mock city and included the following tasks [31]: merging into moving traffic, traffic circle navigation, intersections negotiation, and obstacle avoidance.

Six vehicles out of the eleven finalists managed to complete the 96km long course. To successfully perform the above tasks, an autonomous vehicle requires a suitable set of exteroceptive sensors. The sensors used in the autonomous vehicles of the three best placed teams listed below are used to examine suitable sensor configurations.

1. Tartan Racing's "Boss" [32]
2. Stanford Racing Team's [33]
3. Victor Tango's "Odin" [34]

The listed autonomous vehicles can be seen in Fig. 2.3. A survey of the sensors used in the respective vehicles is given in Tab. 2.1, where the degree of similarity of the sensor configurations is shown to be significant. All autonomous vehicles exhibit monocular video cameras, single-beam, and multi-beam laser scanners, radars (except "Odin"), and a DGPS. It is interesting to note that none of the vehicles use stereo-vision cameras, or 3-D cameras such as a PMD sensor (cf. section 3.4).

a) Tartan Racing's "Boss"

b) Stanford Racing's "Junior"

c) Victor Tango's "Odin"

Figure 2.3: Three winning autonomous vehicles in the DARPA Urban Challenge 2007. Sources: [32–34].

Vehicle	Video	Laser scanner		Radar	DGPS
		Single-beam	Multi-beam		
"Boss" [32]	2 (n/a)	8	1(64), 2(4)	5	1
"Junior" [33]	6 (colour)	2	1(64), 2(4)	2	1
"Odin" [34]	2 (colour)	4	3(4)	0	1

Table 2.1: Sensor configurations used on the three winning autonomous vehicles in the DARPA Urban Challenge 2007. For multi-beam laser scanners the number of beams is given in parentheses.

Besides technological considerations, the costs for the presented sensor systems in Tab. 2.1 is considerable. Still, the use of expensive sensors such as multi-beam laser scanners indicates that truly autonomous driving puts up considerable requirements for exteroceptive sensors. Automotive sensor systems used in series vehicles have to be, and in fact are, much more affordable.

An interesting example for an autonomous vehicle operating without a multi-beam laser scanner is the sensor system of CarOLO's "Caroline" (cf. Fig. 2.4, Rauskolb *et al.* [35]) which placed 7th in the Urban Challenge 2007.

Figure 2.4: CarOLO's "Caroline" competing in the DARPA Urban Challenge 2007. Source: Rauskolb *et al.* [35].

The sensor system shown in Fig. 2.4 is comparable to the sensor systems in Tab. 2.1, with the exception that the 3-D environment map is generated using a stereo camera system as opposed to a multi-beam laser scanner. The sensor system of CarOLO's "Caroline" therefore presents a comparatively inexpensive exteroceptive sensor system while retaining the ability to operate autonomously (cf. Effertz [36]).

2.1.2 Driver Assistance Systems

As opposed to autonomous driving, driver assistance systems merely support the human driver. This results in lower requirements and allows for less expensive sensor configurations. In this section, a number of sensor configurations for driver assistance systems in the literature are discussed. A categorisation of vision sensors for driver assistance systems on intelligent vehicles as proposed by Li and Wang [37] is given in Fig. 2.5.

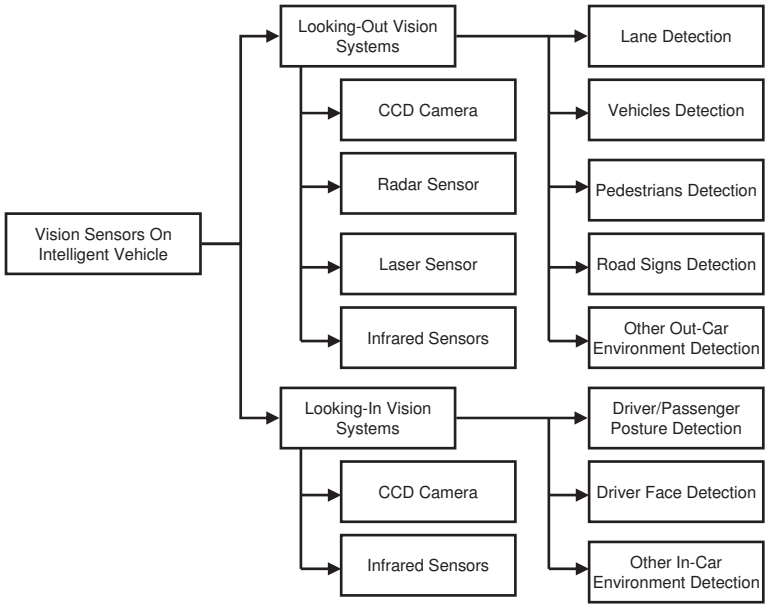


Figure 2.5: Categorisation of looking-out vision systems and looking-in vision systems on intelligent vehicles as proposed by Li and Wang [37]. Source: Li and Wang [37].

The categorisation in Fig. 2.5 distinguishes between looking-out vision systems and looking-in vision systems. For each of these application modes, a number of possible sensors and detection tasks is given. The sensor system of our test vehicle presented in section 1.2.3 resembles the looking-out vision systems in Fig. 2.5 to a remarkable degree. Considering this, looking-in vision systems are excluded here and looking-out vision systems ared discussed in more detail.

Different exteroceptive sensor configurations used in a driver assistance system environment are described in the literature. The test vehicle provided by Audi AG is equipped with a generic multi-modal sensor system that is connected to a central *Automotive Data and Time triggered Framework* (ADTF). From this framework both single-sensor data and fused sensor data can be extracted to design and evaluate future driver assistance systems. A variety of different driver assistance systems and sensor data fusion methods using data

from the ADTF are discussed in the literature such as pedestrian detection by Elias and Mahonen [38] using a PMD camera, parking space exploration by Scheunert *et al.* [39] using a PMD sensor, object tracking by Matzka and Altendorfer [15, 16] using two radar sensors, and car detection by Bergmiller *et al.* [40] using video.

In contrast to autonomous vehicles laser scanners are not used for driver assistance system, thus reducing the costs for sensors considerably. Also most driver assistance system applications use only a single modality as opposed to using a multi-modal representation or a fused environment model. This is partly due to the fact that many driver assistance systems in series vehicles are black-box systems consisting of both sensor and the control unit of the driver assistance system.

2.1.3 Discussion of Automotive Sensor Systems

Considering the sensor systems of different autonomous vehicles a significant resemblance of the sensor systems used on car-like mobile robots and the autonomous cars competing in the DARPA Challenges can be seen. This is in contrast to the limited exteroceptive sensory input of the *VaMoRs* vehicles, using only two video cameras.

The reason for this can in part be found in the real-time constraints of the different systems. Mobile robots do not necessarily operate in real-time, and are thus able to stop, analyse a scene, and then continue with their exploration. This is most obvious when sensors such as a pitching laser scanner are used requiring $4 - 12$ seconds for each scan (cf. Surmann *et al.* [41]), in which the mobile robot, and ideally also the environment, must remain stationary. This assumption does not hold for a road traffic environment.

Even for sensors able to acquire data in real-time, such as video cameras, the amount of data easily exceeds the computational capacity of current series vehicles' electronic control units (ECU). This lead to early autonomous vehicles concentrating on a single modality (i.e. vision) and using only a limited number of cameras at a low resolution. With the considerable increase in computational power and efficient attentional algorithms, this gradually becomes less of a problem.

The sensory system of DARPA vehicles resembles the exteroceptive sensors used on mobile robots, besides the use of ultrasonic range sensors on robots which are replaced by radars in the autonomous cars. Sensor systems used for driver assistance systems can be considered a downscaled version of the autonomous vehicles' sensors. There are two main reasons for this. First, sensors such as multi-beam laser scanners or PTZ video cameras

are still too expensive to be used in current automobiles. Second, the acquired sensor data of a multi-modal sensor system exceeds the computational capacity of current ECUs. This results in most sensor systems relying on a low-resolution video camera and radars to observe the environment. While no truly autonomous driving exists currently with this downscaled sensor system, it is still sufficient for a broad range of driver assistance applications.

Considering this development, we believe that future challenges in the field of driver assistance systems and in autonomous driving will in part rely on active vision systems using existing sensors more efficiently, and to minimise computational requirements at the same time.

2.2 Object Detection and Object Classification

The second integral part of an active vision system is an object detection and classification system. In the following the 2-D object detection and classification method used in our system is presented in section 2.2.1. Section 2.2.2 gives a summary on relevant 3-D object detection and classification methods. A short discussion of the reviewed object detection and classification concepts is given in section 2.2.3.

2.2.1 2-D Object Detection and Classification

Two dimensional object recognition is an active field of investigation that aims to determine semantical information such as location, category, or even identity of an object from visual data, in most cases luminance maps. This process is complex due to four general problems.

First, both scale and shape of objects vary with changes with the position of the object and the observer. Second, the visual appearance of an object is dependent on lighting. Third, the object is subject to occlusion. This can either be a self-occlusion or the object can be occluded by another object that is closer to the observer. Fourth, detection or recognition of an object can be difficult in the presence of distractors referred to as background clutter.

Methods detecting and classifying objects in images and image sequences often rely on machine learning algorithms. According to Bishop [42] the majority of machine learning algorithms can be grouped into unsupervised, semi-supervised, and supervised algorithmic

classes, depending upon the level of human interaction necessary for the training process. In all cases these methods rely on a large set of positive and negative training samples. Computer vision object detection and classification methods can further be categorised into two models: bag-of-word models and part-based models.

In the first approach, a set of small patches (*visual words*) is selected from the training data and stored in a codeword dictionary (cf. Fig. 2.6b). A region is then categorised by observing the frequency of each individual codeword inside the region. Methods using the bag of features approach are proposed by e.g. Sivic *et al.* [43], and Sudderth *et al.* [44]. A general problem for the bag-of-words model is, that the position of the words inside the region is not considered, which is the case for part-based models.

Part-based models consider both the appearance and relative position of object parts for detection and classification (cf. Fig. 2.6c) and were first proposed by Fischler and Elschlager [45]. The main difference between this model and the bag-of-word approach is the additional representation of the connectivity of parts. Current methods using a part-based approach are e.g. Fergus *et al.* [46], and Fei-Fei *et al.* [47].

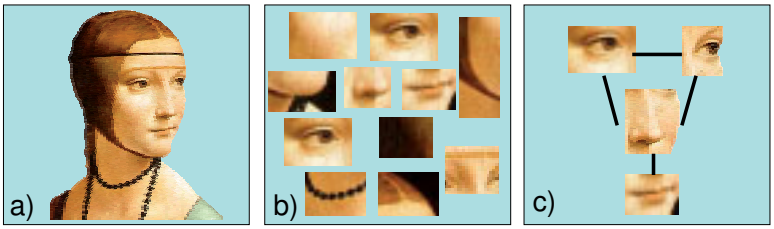


Figure 2.6: Bag-of-words model and part-based model for object detection and classification. The source image a) can be represented in a codeword dictionary b) using the bag-of-words model or using both appearance and relative structure c) in a part-based model. Source: Fei-Fei *et al.* [48, 49].

Apart from using a direct mapping from the source image towards the codebook, current object detection and classification methods other representations such as Haar-like wavelets proposed by Papageorgiou *et al.* [50] or scale invariant features (SIFT) proposed by Lowe [51].

Besides the features used for training and classification, the operational mode, e.g. monolithic algorithms or algorithms structured into several stages is of importance. Given the amount of different object detection and classification algorithms, the further review below is limited to a brief discussion of the Viola and Jones [52] classifier cascade used in

our proposed system, which is also part of the Intel OpenCV image processing library[1] (cf. Bradski and Kaehler [53]). The method proposed by Viola and Jones [52] uses the AdaBoost method proposed by Freund and Schapire [54], which is also described in the following.

The decision of using the Viola and Jones [52] classifier cascade in our proposed system is motivated by the methods's proven efficiency as a robust object detector allowing operation in real-time and AdaBoost's low susceptibility to overfitting [54] as discussed below.

Viola and Jones Face Detector

A method for detection and classification of faces using a boosted cascade of Haar-like features on a video image is proposed by Viola and Jones [52] and has been applied to a large number of object recognition problems since. The method is computationally effective, as it discards most background regions in the first stages of a trained cascade. This allows the algorithm to concentrate its computational resources on regions promising to contain the desired object category.

Cascade of Haar-like Features Motivated by the work presented by Papageorgiou *et al.* [50], a set of rectangular Haar-like features is used by the Viola and Jones face detector. Examples of Haar-like features used in the trained cascades are given in Fig. 2.7.

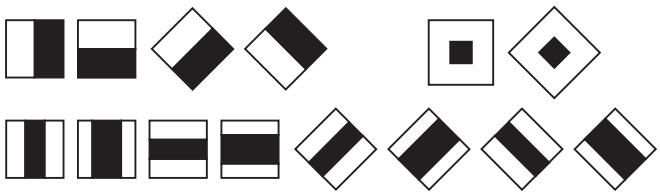


Figure 2.7: Haar-like features used in the trained cascades are edge features (top left), centre-surround features (top right), and line features (bottom) Source: Viola and Jones [52].

The cascade is built by iteratively adding simple Haar-like features to a stage in the cascade until it rejects a certain fraction (e.g. 0.500) of negative samples remaining after the previous stages. At the same time, each stage in the cascade is constrained to reject no, or only a very small fraction (e.g. 0.003) of positive samples. An example feature

[1] Available online at http://sourceforge.net/projects/opencvlibrary/

value distribution can be seen in Fig. 2.8. In the literature, the number of both positive and negative samples used for training usually exceeds 10^3 samples for each group.

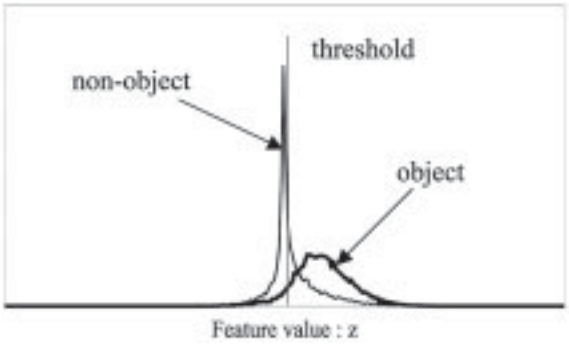


Figure 2.8: Example of feature value distributions. The Viola and Jones face detector [52] iteratively selects a single feature as a weak classifier that best separates the two classes with a threshold (blue). Source: Mita *et al.* [55].

Integral Image Haar-like features can be rescaled easily which is also exploited by Viola and Jones [52], where the integral image representation is used. The concept of using an integral image extends the summed area table proposed by Crow [56]. Inside the summed area table or integral image, each pixel $sat(i, j)$ stores the sum of all pixels within a rectangular region towards the upper-left corner. The integral image can be calculated in a single pass using Eq. 2.1.

$$sat(i, j) = sat(i - 1, j) + \sum_{i' \leq i} l(i', j) \tag{2.1}$$

The sum of pixel values within any rectangle inside the image can now be calculated within constant time by using four array references

$$\sum_{(i,j) \in D} l(i, j) = sat(D_{UL}) + sat(D_{LR}) - sat(D_{UR}) - sat(D_{LL}) \tag{2.2}$$

where D is the area the sum of pixel values is to be determined for, and D_{UL}, D_{UR}, D_{LL}, D_{LR} denote upper-left, upper-right, lower-left, and lower-right corners of D respectively.

In Fig. 2.9 the integral image for an example source image is shown. There the sum of all pixels inside the red rectangle can be determined to be

$$\sum_{(i,j) \in D} l(i, j) = (1.88 + 5.17 - 3.41 - 2.92) \cdot 10^6 = 7.22 \cdot 10^5$$

in constant time using Eq. 2.2.

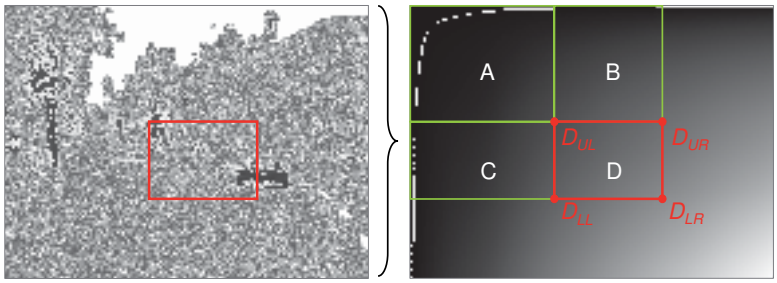


Figure 2.9: Integral image concept proposed by Crow [56]. The value of the integral image at location D_{UL} equals the sum of the pixels in A. At D_{UR} this value is A+B, at D_{LL} it is A+C. Computing the pixel value sum within D is done by $D_{UL} + D_{LR} - D_{UR} - D_{LL}$.

AdaBoost

Adaptive Boosting (AdaBoost), is a machine learning algorithm, presented by Freund and Schapire [54] and improved using confidence-rated predictions by Schapire and Singer [57]. AdaBoost is adaptive as subsequent classifiers are tuned towards correctly classifying data that was misclassified by previous classifiers. Freund and Schapire [54] claim that AdaBoost is sensitive towards noisy data and outliers, yet less susceptible to overfitting than most other machine learning algorithms.

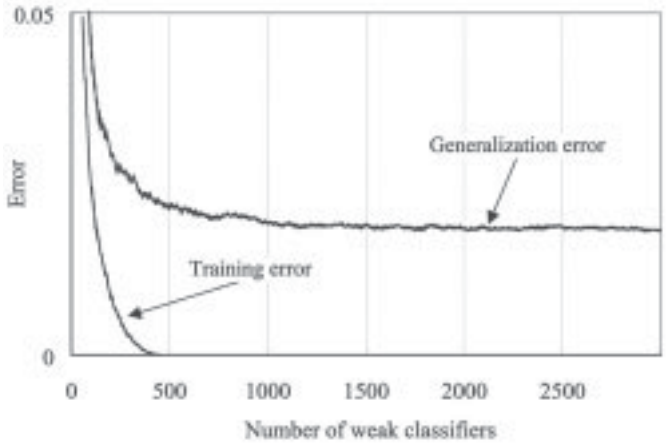


Figure 2.10: Overfitting using the Viola and Jones face detector [52]. While the training error converges to zero, the generalisation error is no longer reduced after 10^3 features are selected. Source: Mita *et al.* [55].

Overfitting describes an effect observed for supervised generation of a statistical model with too many parameters. If more parameters are used, the performance on the training-set increases, but does not decrease the error on a test-set. This implies that no additional discriminative features exist and no further improvement can be expected by adding more

features. For the face detector presented by Viola and Jones [52] using AdaBoost this effect is apparent if more than 10^3 features are used according to Mita *et al.* [55] (cf. Fig. 2.10).

Operation of AdaBoost AdaBoost determines a set of weak classifiers in a total of N rounds. In each round n, the weight distribution for the individual samples in the data D_n is updated. Previously misclassified samples are increased in weight, whereas correctly classified samples' weights are decreased. This causes the new classifier to regard the previously misclassified samples more than those correctly classified.

A strong classifier $H(x)$ is determined using Alg. 2.1 as described by Freund and Schapire [54].

Input: Sample vector x, class label vector y, and sample weight vector D, all of length m, Number of rounds N
Output: Strong classifier H
for *($i \leftarrow 1$* **to** m*)* **do**
  $D_1(i) = 1/m$
end
for *($n \leftarrow 1$* **to** N*)* **do**
  repeat
    $\epsilon_j = 0$;
    Choose weak classifier $h_n : X \rightarrow \{-1, +1\}$;
    for *($i \leftarrow 1$* **to** m*)* **do**
      if $y_i \neq h_j(x_i)$ **then**
        $\epsilon_j = \epsilon_j + D_n(i)$;
      end
    end
    ϵ_n = weighted error rate of h_n;
    if $\epsilon_n < 0.5$ **then**
      $\alpha_n = 0.5 \ln((1 - \epsilon_n)/(\epsilon_n))$;
      for *($i \leftarrow 1$* **to** m*)* **do**
        $D_{n+1}(i) = D_n(i)\, e^{-\alpha_n y_i h_n(x_i)}$;
        normalise $D_{n+1}(i)$;
      end
    end
  until ϵ_j *is minimised* ;
end
$H(x) = 0$;
for *($n \leftarrow 1$* **to** N*)* **do**
  $H(x) = H(x) + \alpha_n h_n(x)$;
end
H(x) = sign(H(x));

Algorithm 2.1: AdaBoost machine learning algorithm as described by Freund and Schapire [54].

2.2.2 3-D Object Detection and Classification

Apart from detecting and classifying images on 2-D sensor data, 3-D sensors are able to observe the environment using a different modality and therefore able to detect objects that cannot be detected by 2-D sensors (cf. Fig. 7.13).

Examples for 3-D sensors are photonic mixer device (PMD) cameras as desrcibed in Fardi *et al.* [58], or binocular stereo cameras, e.g. Cochran and Medioni [59]. Multi-beam laser scanners to classify and track traffic participants are also described in the literature, e.g. Gidel *et al.* [60]. Besides these dedicated 3-D sensors, numerous attempts have been made to extract 3-D information from 2-D sensors. Examples are *structure from motion* (e.g. Chiuso *et al.* [61], Brostow *et al.* [62]), and *shape from defocus* (e.g. Favaro and Soatto [63]) for monocular cameras, or using a pitching motion to extract 3-D data from a single-beam laser scanner (e.g. Ryde and Hu [64]).

Once 3-D information about the environment is acquired, characterising the object surface is a commonly used technique for detection and classification. There exist a number of surface descriptors, four of which are discussed below. The object descriptors are then compared to descriptor sets in an object database using a similarity measure to detect and classify an object.

Fundamental Surface Types and Shape Index

Besl and Jain [65] propose to differentiate between eight fundamental types of surfaces using Gaussian curvature K and mean curvature H. As both values can be negative, zero, or positive, a total of $3^2 - 1 = 8$ different combinations are possible (the combination $K < 0$ and $H = 0$ is not possible). These represent different surface types, namely (sorted from concave to convex): pit, valley, saddle valley, flat, saddle ridge, ridge, and peak. The eighth surface type is called a minimal surface and is a surface with a mean curvature of zero. Generally, only almost flat surfaces have that property. However there are certain non-flat surfaces, such as a the inside of some tori, which are owning that property as well. Examples for the different surface types can be seen in Fig. 2.11.

Closely related to fundamental surface descriptors, the use of a shape index is proposed by Koenderink and van Doorn [66], which can be thought of as a continuous description of the fundamental surfaces in Besl and Jain [65]. This idea is also used for generic classification of 3-D objects by Csákány and Wallace [67]. There, neighbouring points

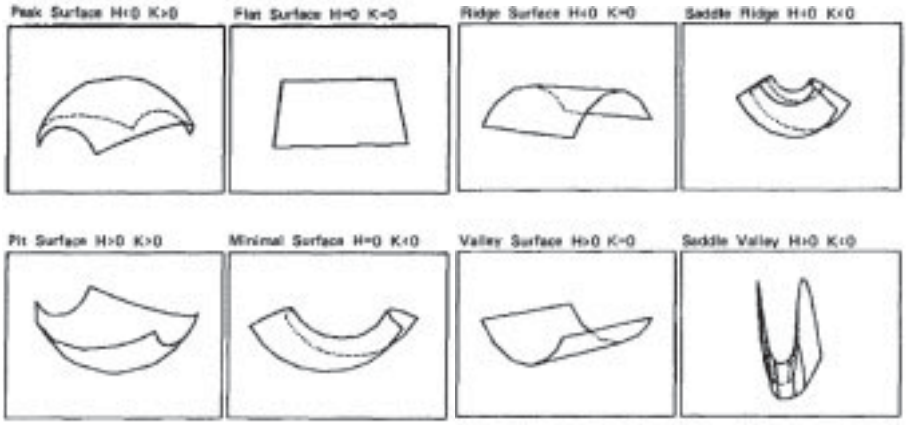
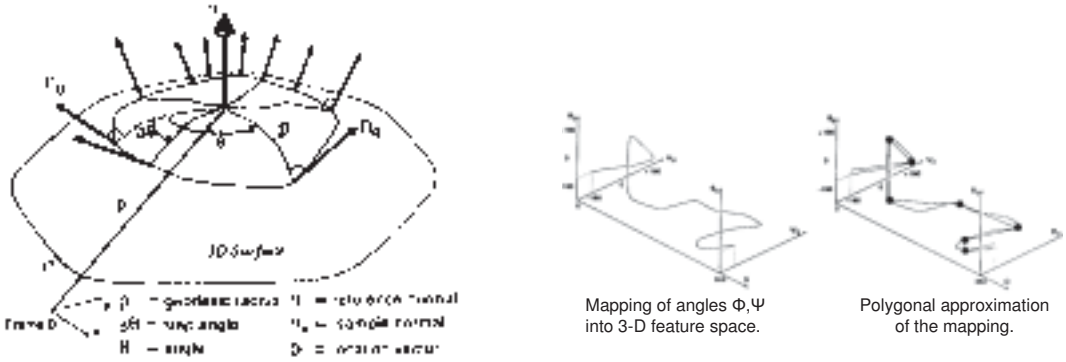


Figure 2.11: Eight fundamental surface types as proposed by Besl and Jain [65] differentiated using Gaussian curvature K and mean curvature H. Source: Besl and Jain [65].

with the same associated surface label are grouped to form surface patches. The likelihood of an object to be classified as a member of a certain class is then based on statistical indicators accounting for the likelihood and cardinality of the feature set associated with a certain class.

Splash Image

The splash image method is proposed by Stein and Medioni [68]. As a means to avoid the use of second derivatives susceptible to noise, splash images are calculated using a circle of normal vectors around the surface normal of a certain point. An illustration of this technique is given in Fig. 2.12a.



a) b)

Figure 2.12: a) Splash image calculation on a 3-D surface using a circle of normal vectors. b) Mapping of angular deviations into a 3-D feature space (left) and its polygonal approximation (right). Source: Stein and Medioni [68].

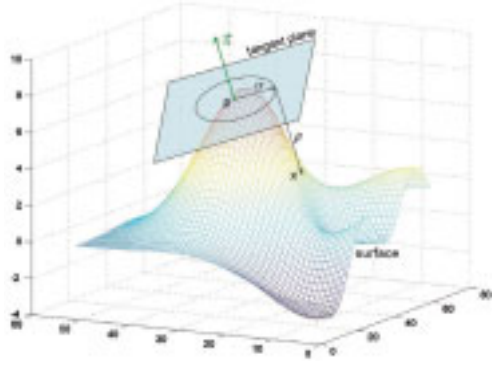
After determining the surrounding surface normals, the angular deviations of the surrounding surface normals relative to the central surface normal are mapped into a 3-D feature space, as shown in Fig. 2.12b. There, a polygonal approximation of the original mapping is shown as well, allowing for a parametric representation of the mapping. These parameters are used as key features when searching for a suitable match in the object database.

Spin Image

A shape-based 3-D classification algorithm for classification of multiple objects in scenes containing clutter and occlusion is presented in Johnson and Hebert [69]. Spin images are a shape descriptor working at the data level. The methods' classification performance is evaluated in Johnson and Hebert [69] and shown to be robust.

Spin images use an object-centred coordinate system, where surfaces are compared by matching individual surface points as opposed to complete surfaces. The problem of matching a complete surface is thus divided into the problem of matching a number of surface points, thereby reducing complexity. It is argued in Johnson and Hebert [69] that clutter points do not have a correspondence on a nearby surface, while partly occluded surfaces in the scene do not affect the classification, as they are rejected.

Each point is represented by a 2-D spin image, which is created by virtually spinning a plane around an oriented point, a 3-D point with an associated surface normal. Relative to this point, two coordinates are defined: perpendicular distance to the surface normal α and perpendicular distance to the tangent plane on the oriented point β (cf. Fig. 2.13a).

a) 

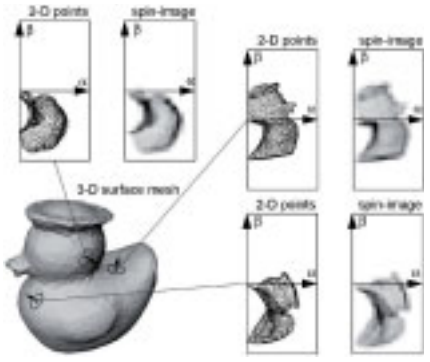
b) 

Figure 2.13: Figure a) shows an oriented point p on a 3-D surface with the point's surface normal $\vec{n}$ and the corresponding tangent plane. Figure b) shows spin images for three oriented points on a 3-D surface mesh of a rubber duck. Source: Johnson and Hebert [69].

A 2-D accumulator field indexed by α and β is then created. The axes' lengths correspond to the support distance of the spin image. The respective indexes α and β are calculated for all points on the surface within support distance and the spin image bin at (α, β) is incremented using an interpolation method.

If bin values are seen as intensity values, the accumulator field can be seen as an image, which reduces the problem to 2-D object detection and classification as discussed in section 2.2.1.

Spin Image Generation Spin image generation can be thought of as spinning a plane around the surface normal $\vec{n}$ of an oriented point p, while binning all surface points x as they intersect the plane. Figure 2.13a shows a 3-D surface, with an oriented point p, as well as its tangent plane and the surface normal $\vec{n}$.

The two values α and β are used as bin indexes in the spin image and represent the perpendicular distance to the surface normal through the oriented point (α) and the perpendicular distance to the tangent plane (β) respectively.

$$\beta = \quad \frac{\vec{n}}{||\vec{n}||} \cdot (\vec{x} - \vec{p}) \tag{2.3}$$

$$\alpha = \quad \sqrt{\|\vec{x} - \vec{p}\|^2 - \beta^2} \tag{2.4}$$

Calculation of α and β is then performed for every measured surface point x. For the real values gained, an interpolation method is used for the contribution to discrete bins.

Spin Image Matching A correlation coefficient indicates the strength and relation of a linear relationship between two variables. Spin images are matched towards a database using the Pearson product-moment correlation coefficient, which is obtained by dividing the covariance of the two variables by the product of their standard deviations. The correlation coefficient $R(P, Q)$ between two points P and Q in two spin images can be determined with Eq. 2.5.

$$R(P,Q) = \frac{N \sum P_i Q_i - \sum P_i \sum Q_i}{\sqrt{(N \sum P_i^2 - (\sum P_i)^2)(N \sum Q_i^2 - (\sum Q_i)^2)}} \tag{2.5}$$

The correlation coefficient R will take values from $R = -1$ (entirely anti-correlated)

to $R = 1$ (entirely correlated). R can thus be used to asses the similarity of two images. If R is high, the spin images are similar, for small or negative R values, the spin images are considered not similar.

Comparison of Surface Descriptors

A summary of 3-D data surface descriptor techniques is presented by Campbell and Flynn [70]. It lists a multitude of methods as well as their ability to handle local occlusion, database size, recognition rate, and their respective complexity. There, the recognition rate ranges from 0.77 to 1.00 with database sizes of 4–48 object classes. Of these, both the point signature approach proposed by Chua and Jarvis [71] and the spin image method proposed by Johnson and Hebert [69] show a robust classification performance with a recognition rate of 1.00 in the presence of clutter.

2.2.3 Discussion of Object Detection and Object Classification

In sections 2.2.1 and 2.2.2 a number of object detection and classification methods to be used on 2-D data and or 3-D data are presented.

In the field of 2-D object detection and classification this thesis focuses on the Viola and Jones face detector using Haar-like features as simple cues, due to the computational efficiency of the cascaded approach and the shown performance in a number of applications. For 3-D object detection and classification a selection of surface representation techniques that can be used to detect and to classify an object is presented as well as a comparison of 3-D surface descriptors by Campbell and Flynn [70].

Both 2-D and 3-D object detection and classification are active fields of research, with a number of well tested algorithms that can be used for training the classifiers, and performing object detection and classification. The Viola and Jones face detector in particular is a widely used and proven method for use in real-time systems.

For an application in an automotive vision system, one requirement is the ability to perform in real-time, a constraint that can be fulfilled by few algorithms including the Viola and Jones classifier cascades. The field of 3-D object detection and classification is promising considering the additional information gained, but must be significantly down-scaled and enhanced in robustness for use in series vehicles.

2.3 Decision Making

The third integral part of an active vision system is the decision making process required to select the best regions to be observed from the set of candidate regions. In the literature decision making has been defined as

> *"a reasoning process that leads towards the selection of one alternative over others."* (Reason [72]).

In our case this selection can lead to serious consequences if a critical traffic participant is overseen. For this, besides considering possible approaches to make decision in a computer system, the actual permissibility of making a decision must be considered first. Therefore, a short overview on moral theories and ethical limitations for decision making under risk is given in in section 2.3.1. In section 2.3.2 the concept of Pareto optimal decision making is presented. Utility functions to determine the overall utility of a solution are discussed in section 2.3.3. A short review of multi-agent resource allocation is given section 2.3.4. Finally the presented decision making concepts are discussed in section 2.3.5.

2.3.1 Moral Theories on Risk

Risk, in the sense that one or more traffic participants can be injured or even killed if they are not recognised in time, is treated from a utilitarian viewpoint in our proposed system. An example for a utilitarian approach is the preference for an event with small overall severity of injuries over an event with a higher overall severity of injuries. This does not imply that this is the best, or the only way that risk can be treated in a decision making system.

Decision Making under Risk

Our proposed system's goal stated in section 1.2.2 is to make decisions that minimise the negative impact on the environment, foremost the safety of all traffic participants. Decision making necessitates the deliberation of individuals' risks. The dangers of postponing the observation of a pedestrian have to be weighted against the safety gained by prioritising a bicyclist. This class of problems is located in the field of moral philosophy or applied ethics. In practice however, the problems of risk are rarely treated in moral philosophy, as

> *"... the problem of appraising risks from a moral viewpoint does not seem to have any satisfactory solution in established moral theories."* (Zalta [73])

This lack of solutions is not considered to be caused by a lack of established moral theories, but rather the ineptness of these for the problem of appraising risks. Examples for moral theories on risk are utilitarianism presented by von Neumann and Morgenstern [74], rights-based moralities proposed by Nozick [75], and contract theories presented by Scanlon [76].

Utilitarianism is a neutral decision strategy, where the best solution to a problem involving risk is coincident with the statistically optimal solution. From a utilitarian standpoint a possibly disastrous event with a very small probability of happening can override a considerable probability of hurting a single individual. In Zalta [73] the example of the preference of one person being inevitably injured ($1 \times 1.00 = 1.00$) against the probability of 1000 people being injured with an individual probability of 0.0011 ($1000 \times 0.0011 = 1.10$) demonstrates this problem.

Rights-based moral theory presented by Nozick [75] argues that no person has the right to injure another person, which implies that no person has the right to increase the probability that another person is injured. Strictly interpreted, this will effectively deny any person the right to operate a car due to the increase in risk of other people to be injured.

Contract theory proposed by Scanlon [76] is able to resolve some of the problems of rights-based moral theory in so far as it would enable people to consent to motorised road traffic even if this increases the probability of being injured. Such a consent would have to be unanimous, or otherwise any person could deny any other person's right to operate a car. This option will eventually be as impracticable as the rights-based moral approach.

In actual societies, the problem of granting the right of operating a vehicle is made socially acceptable by allowing for a reciprocal exchange of the risks and benefits of participating in traffic. Any person is allowed to participate in road traffic – and thus increase the probability of other people to be injured – by granting the same right to any other person.

Decision Making in Law

Ius in Bello Imposing severe risks upon other people's lives constitutes a sad reality in warfare. The achievement of a certain, however justified, goal is subject to conventions, such as the Fourth Geneva Convention [77]. There, non-combatants, or civilians, are granted immunity. This immunity is granted with respect to the fact that civilians do not carry weapons, and therefore do not constitute a risk to an enemy soldier (cf. Walzer [78]).

The asymmetry of threat between a civilian and an enemy soldier finds its analogy in a pedestrian, or a bicycle on the one side, and a motorised traffic participant such as a motorcycle, car, or lorry on the other side. While any motorised traffic participant can present a lethal risk to an unmotorised traffic participant, the same cannot be said vice versa. Originating from this asymmetry, a different treatment of unmotorised traffic participants appears to be necessary.

In Time of Peace Besides international conventions applicable during times of war, common law systems regulate the permissibility of endangering peoples' life and physical integrity. Using the European Convention on Human Rights [79] as one representative system, a person's life is its most precious asset, and therefore not negotiable. There is no most-favoured treatment, so that negotiating the life of a single person against the life and physical integrity of two or more people is considered unlawful.

Doctrine of Double Effect Considering the moral problems and judicial positions towards decision making under risk, the doctrine of double effect is sometimes used to explain the

> *"... permissibility of an action that causes a serious harm, such as the death of a human being, as a side effect of promoting some good end."* (Zalta [80])

The original thought is credited to Aquinas [81] and provides four conditions under which the double effect can be considered to be (cf. Zalta [80])

- The action in itself from its very object is good or at least indifferent.
- The good effect and not the evil effect is intended (cf. Mangan [82]).
- The good effect is not produced by means of the evil effect.
- There is a proportionately grave reason for permitting the evil effect.

The above conditions can be found pervasively in many fields of moral theory, but prove difficult to be applied as such considering certain ethical limitations discussed below.

Ethical Limitations of Decision Making

There exist multiple moral theories on risk, that must be considered in a decision making system. However, these theories fail to provide a satisfactory solution to the problems connected with decisions on risk. In our example of postponing the observation of a pedestrian to prioritise a bicyclist, decision making in human drivers is dependent on a subjective rationality. This rationality shows cultural influences, such as the status of different traffic participants.

Decision making in an active vision system necessitates to relate different traffic participants to possible injuries caused by accidents. This includes to relate different injury severities to each other for different traffic participants. Present driver assistance systems such as pedestrian detection systems avoid to relate different traffic participants against each other by design, detecting only a single traffic participant class. In order to include different traffic participants, a utilitarian system recommends itself due to its ability to quantify these relations and thus make the system operable.

As long as the driver assistance system does not operate the vehicle autonomously, a utilitarian decision making instance provides additional information and thus additional safety. For autonomous driving systems, this does not hold as the utilitarian logic is often adverse to a commonly accepted subjective rational standpoint. At this point, the use of a utilitarian system must be reconsidered. It is possible to adapt and extend the utility functions used in the system to reflect a society's values and morals, such as the special protection of pedestrians. This process of deliberation must be transparent to each member of society and has to be conducted by a democratically legitimate body. Only then can the reciprocal exchange of the risks and benefits of an extended utilitarian decision making system be considered morally acceptable.

2.3.2 Pareto Efficiency

A decision making process involved in determining the optimum resource allocation is a multiobjective optimisation problem. According to Chevaleyre *et al.* [83] any acceptable solution for this class of problems is necessarily Pareto efficient. The set of Pareto efficient

solutions is called a Pareto frontier and connects all solutions that are not dominated by any other solution. A solution is said to be dominated by another solution if there is at least one other possible solution which shows an increase in one objective while exhibiting the same or better results for all other objectives (cf. Fig. 2.14). All solutions on the Pareto frontier are optimal in an objective sense, transforming the problem towards the selection of only one optimum solution, which is inherently subjective.

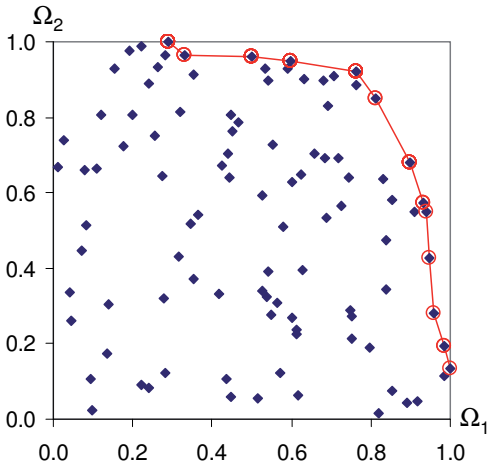


Figure 2.14: Pareto frontier for two objective dimensions Ω_1 and Ω_2 connecting 13 utility tuples (red) determined from 100 utility tuples (blue). Higher values for Ω_1 and Ω_2 are considered better.

It is not necessary to determine the Pareto frontier in the decision making process but it is necessary to ensure that every solution produced by the decision making process is an element of the Pareto frontier and thereby Pareto efficient.

2.3.3 Multiobjective Resource Allocation

Multiobjective resource allocation is a multicriteria decision making process for the assignment (eg. Triantaphyllou [84], Ehrgott [85]) and scheduling (e.g. T'kind and Billaut [86]) of resources. In the field of multiobjective resource allocation a range of algorithms to compare solutions using multiple properties exist, such as

- Utilitarian (maximise sum of utilities or minimising sum of utility losses)
- Egalitarian (minimise worst utility loss or maximise least utility gain)
- Elitist (maximise highest utility)
- Nash product (maximise utility product)
- Leximin ordering (sort by utility losses in ascending order)

where utility is a quantity determined by the utility function.

The notion of combined utility can be used as a measure to determine the overall quality of a single allocation as well as, considering a large number of allocations, the quality of an allocation mechanism. Methods to ensure optimum combined utility differ widely, however there are two main properties to consider: *efficiency* and *fairness*. Besides the concept of Pareto efficiency, which is discussed above, the concept of fairness requires to ensure that the chosen allocation must be beneficial for as many individual objectives as possible.

In this thesis a Pareto efficient decision making concept is proposed to allocate resources. For this, the different utility algorithms are evaluated using both synthetic data and real data acquired using test sequences.

The expected utility $\mathcal{U}(\mathcal{R}_m)$ of observing a region $\mathcal{R}_m$ can be calculated using the set of objectives $\Omega_{1,..,N}$. Below, several utility functions are discussed, which can serve as a measure for global utility. As exemplary utility functions for two objectives $\Omega_{1,2}$ evaluating three candidate regions $\mathcal{R}_{1,2,3}$, the following table of utilities $\mathcal{U}_n(\mathcal{R}_m)$ is assumed:

Utility	Ω_1	Ω_2	$\longrightarrow$	Utility loss	Ω_1	Ω_2
$\mathcal{R}_1$	0	3		$\mathcal{R}_1$	5	0
$\mathcal{R}_2$	3	2		$\mathcal{R}_2$	2	1
$\mathcal{R}_3$	5	1		$\mathcal{R}_3$	0	2

Table 2.2: Example utility map for two objectives $\Omega_{1,2}$ evaluating three candidate regions $\mathcal{R}_{1,2,3}$. Utility loss is determined as the difference to the highest utility value for the same $(\Omega, \mathcal{R})$ combination.

Table 2.2 states the utilities $\mathcal{U}_n$ for individual regions and objectives but also the utility losses. Utility loss (henceforth referred to as $\mathcal{U}^{\downarrow}$) is the utility difference between the candidate region with the highest utility and the considered candidate region.

$$\mathcal{U}_n^{\downarrow}(\mathcal{R}_m) = \max_{\iota=1,..,N_{\mathcal{R}}} (\mathcal{U}_n(\mathcal{R}_\iota)) \; - \; \mathcal{U}_n(\mathcal{R}_m) \tag{2.6}$$

Utilitarian Utility

Utilitarian utility $\mathcal{U}^u(\mathcal{R}_m)$ for a region $\mathcal{R}_m$ is defined to be the sum of all objectives' individual utilities $\mathcal{U}_n(\mathcal{R}_m)$ and is calculated using Eq. 2.7.

$$\mathcal{U}^u(\mathcal{R}_m) = \sum_{n=1,..,N_\Omega} \mathcal{U}_n(\mathcal{R}_m) \tag{2.7}$$

$$\mathcal{U}^u(\mathcal{R}_m)^* = \arg\max_m \mathcal{U}^u(\mathcal{R}_m) \tag{2.8}$$

Using the utility distribution from Tab. 2.2, candidate region $\mathcal{R}_3$ exhibits the highest combined utility $\mathcal{U}^u(\mathcal{R}_m)^*$, as

$$\mathcal{U}^u(\mathcal{R}_m)^* = \mathcal{U}^u(\mathcal{R}_3) = 5 + 1 = 6$$

While a utilitarian resource allocation ensures overall high local utilities, it cannot ensure fairness.

Nash product Utility

The Nash product utility $\mathcal{U}^\times(\mathcal{R}_m)$ for a region $\mathcal{R}_m$ is defined as the product of the objectives' utilities (cf. Eq. 2.9). It derives its name from non-cooperative game theory by Nash [87].

$$\mathcal{U}^\times(\mathcal{R}_m) = \prod_{n=1,..,N_\Omega} \mathcal{U}_n(\mathcal{R}_m) \tag{2.9}$$

$$\mathcal{U}^\times(\mathcal{R}_m)^* = \arg\max_m \mathcal{U}^\times(\mathcal{R}_m) \tag{2.10}$$

Assuming all utility values to be positive, the Nash product favours increases in overall utility, but also inequality-reducing redistributions. Therefore, using the utility distribution from Tab. 2.2, the highest global utility is be gained for region $\mathcal{R}_2$.

$$\mathcal{U}^\times(\mathcal{R}_m)^* = \mathcal{U}^\times(\mathcal{R}_2) = 3 \times 2 = 6$$

Using a Nash product allocation is useful as it supports a balanced set of high local utilities. An obvious problem is the behaviour if at least one objective states a utility of

zero or below, which results in the annihilation or sign change of the combined utility for that region. While this is generally desirable to maintain fairness, it offers a high leverage for a single objective which is problematic. An offset value is commonly added to mitigate this problem.

Egalitarian Utility

Egalitarian utility is determined by the objective fulfilment currently worst off. This term is ambivalent, as 'worst off' could either indicate that the utility of an objective is small, or that the utility loss is high. For the first case $\mathcal{U}^e$ the minimum utility for any objective maximised (cf. Eq. 2.11, 2.12),

$$\mathcal{U}^e(\mathcal{R}_m) = \min_{n=1,..,N_\Omega} \mathcal{U}_n(\mathcal{R}_m) \tag{2.11}$$

$$\mathcal{U}^e(\mathcal{R}_m)^* = \arg\max_m \mathcal{U}^e(\mathcal{R}_m) \tag{2.12}$$

for the second case $\mathcal{U}^{e\downarrow}$ the maximum utility loss $\mathcal{U}^\downarrow$ is minimised (cf. Eq. 2.13, 2.14).

$$\mathcal{U}^{e\downarrow}(\mathcal{R}_m) = \max_{n=1,..,N_\Omega} \mathcal{U}_n^\downarrow(\mathcal{R}_m) \tag{2.13}$$

$$\mathcal{U}^{e\downarrow}(\mathcal{R}_m)^* = \arg\min_m \mathcal{U}^{e\downarrow}(\mathcal{R}_m) \tag{2.14}$$

Using the utility distribution from Tab. 2.2, the optimum utilities are

$$\mathcal{U}^e(\mathcal{R}_m)^* = \mathcal{U}^e(\mathcal{R}_2) = \min(3;2) = 2$$

for $\mathcal{U}^e$, and

$$\mathcal{U}^{e\downarrow}(\mathcal{R}_m)^* = \begin{cases} \mathcal{U}^{e\downarrow}(\mathcal{R}_2) = \max(2;1) = 2 \\ \mathcal{U}^{e\downarrow}(\mathcal{R}_3) = \max(0;2) = 2 \end{cases}$$

for $\mathcal{U}^{e\downarrow}$.

Egalitarian resource allocations ensures that every objective is considered at the expense of optimising towards a high overall utility. As only the single worst-off objective is considered, a solution found with an egalitarian utility function is not necessarily Pareto efficient. Pareto efficiency for an egalitarian utility concept can be attained by using

Leximin ordering presented below.

Elitist Utility

Elitist utility $\mathcal{U}^{\$}(\mathcal{R}_m)$ is governed by the objective currently best off (cf. Eq. 2.15, 2.16) and therefore is diametrically opposed to the egalitarian utility function discussed above. It can easily be seen that it is not a fair utility concept nor does it ensure Pareto efficiency.

$$\mathcal{U}^{\$}(\mathcal{R}_m) = \max_{n=1,..,N_\Omega} \mathcal{U}_n^{\$}(\mathcal{R}_m) \tag{2.15}$$

$$\mathcal{U}^{\$}(\mathcal{R}_m)^* = \arg\max_m \mathcal{U}^{\$}(\mathcal{R}_m) \tag{2.16}$$

Using the utility distribution from Tab. 2.2, the elitist utility allocation results in

$$\mathcal{U}^{\$}(\mathcal{R}_m)^* = \mathcal{U}^{\$}(\mathcal{R}_3) = \max(5;1) = 5$$

Leximin Ordering Utility

Leximin ordering utility $\mathcal{U}^{\lambda}$ and $\mathcal{U}^{\lambda\downarrow}$ can be seen as a refinement of egalitarian utility. This method does not only consider the utility gained by the worst alternative but an ordered utility vector from all alternatives. All objectives' utilities or utility losses for each candidate region $\mathcal{R}$ are sorted in ascending order inside an ordered vector for $\mathcal{U}^{\lambda}$, or descending order for $\mathcal{U}^{\lambda\downarrow}$ respectively. To determine the best utility vector, the first elements of each vector are compared against each other. If more than one optimal solution (i.e. maximum utility or minimum utility loss) is found, then the second, third, etc. elements of all remaining optimum solution vector are compared to each other.

In our example, candidate regions $\mathcal{R}_{1,2,3}$ are assigned the following ordered utility vectors

$$\mathcal{U}^{\lambda}(\mathcal{R}_2) = (2;3) \;>\; \mathcal{U}^{\lambda}(\mathcal{R}_3) = (1;5) \;>\; \mathcal{U}^{\lambda}(\mathcal{R}_1) = (0;3)$$

and ordered utility loss vectors

$$\mathcal{U}^{\lambda\downarrow}(\mathcal{R}_3) = (2;0) \;<\; \mathcal{U}^{\lambda\downarrow}(\mathcal{R}_2) = (2;1) \;<\; \mathcal{U}^{\lambda\downarrow}(\mathcal{R}_1) = (5;0)$$

The optimal allocations using leximin ordering for the utility distribution given in Tab. 2.2 are

$$\mathcal{U}^{\lambda}(\mathcal{R}_m)^* = \mathcal{U}^{\lambda}(\mathcal{R}_2) = (2;3)$$

for $\mathcal{U}^{\lambda}$, and

$$\mathcal{U}^{\lambda\downarrow}(\mathcal{R}_m)^* = \mathcal{U}^{\lambda\downarrow}(\mathcal{R}_3) = (2;0)$$

for $\mathcal{U}^{\lambda\downarrow}$.

The problem of identical egalitarian utility for $\mathcal{U}^{e\downarrow}(\mathcal{R}_2) = \mathcal{U}^{e\downarrow}(\mathcal{R}_3) = 2$ is resolved using leximin ordering, establishing a preference of $\mathcal{U}^{\lambda\downarrow}(\mathcal{R}_3) = (2;0)$ over $\mathcal{U}^{\lambda\downarrow}(\mathcal{R}_2) = (2;1)$.

Summary of Region Preferences

As expected, different utility concepts result in different region preferences. In Tab. 2.3 the above results are summarised for an overview.

Region	Ω_1	Ω_2	Concept
$\mathcal{R}_1$	0	3	–
$\mathcal{R}_2$	3	2	$\mathcal{U}^{e}$,$\mathcal{U}^{e\downarrow}$,$\mathcal{U}^{\times}$,$\mathcal{U}^{\lambda}$
$\mathcal{R}_3$	5	1	$\mathcal{U}^{u}$,$\mathcal{U}^{e\downarrow}$,$\mathcal{U}^{\$}$,$\mathcal{U}^{\lambda\downarrow}$

Table 2.3: Region preferences for example regions and objectives.

It can be seen from Tab. 2.3, that different utility concepts result in different optimum allocations. In order to determine the optimum allocation in an active vision system, either a single utility concept, or multiple utility concepts can be used. Both approaches are described below.

Utility combination Concepts

In the literature, both single algorithm approaches for decision making and the use of multiple algorithms is proposed. A summary of both approaches is given in the following.

Single Algorithm Approach In a single algorithm approach, only a single decision making method is used to determine the regions' utilities. A schematic overview for both single algorithm approaches and multiple algorithm approaches for objective combination is given in Fig. 2.15.

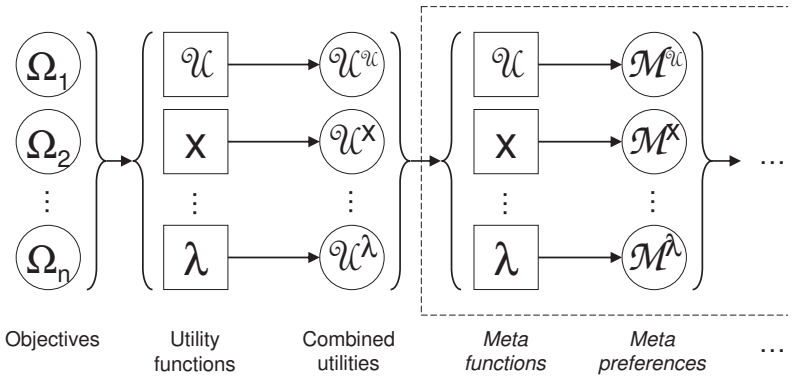


Figure 2.15: Single algorithm and multiple algorithm approaches for objective combination. Whereas a single algorithm approach uses only a single combined utility $\mathcal{U}$ value to determine the optimum solution, a multiple algorithm approach uses at least one meta layer to determine a meta preference $\mathcal{M}$.

Multiple Algorithms Approach As an alternative to a single algorithm approach, concepts with multiple utility algorithms are proposed in the literature. For example, Seara and Schmidt [88, 89] use a weighted voting method in a winner selection society to determine the best region to be observed. The winner selection society consists of different agents representing different utility algorithms.

In theory, multiple algorithm approaches are preferable due to the increased level of robustness considering different combined utility concepts. In practice, this holds if the different utility algorithms generally result in the same region preferences. However, in case of dissenting region preferences the problem of selecting the best solution from a set of objectives is effectively transformed into selecting the best solution from a set of utility algorithm preferences. This meta preference can again be determined using a single algorithm or multiple algorithm approach. Note that this can incur an infinite loop of expressing meta-preferences using meta functions.

2.3.4 Multiagent Resource Allocation

Besides centralised multiobjective algorithms discussed in section 2.3.3, or evolutionary algorithms (e.g. Coello *et al.* [90]), resource allocation using multiagent resource allocation is discussed in Chevaleyre *et al.* [83] with an emphasis on a formal description of the problem and social welfare metrics.

The nature of an agent has been defined differently in past publications. Franklin and Graesser [91] examine various definitions of the term *agent* and suggest that

> *"... an autonomous agent is a system situated within and a part of an environment that senses the environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future."* (Franklin and Graesser [91])

In Wooldridge [92] the following, broader, definition adapted from Wooldridge and Jennings [93] is given:

> *"An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment to meet its design objectives."* (Wooldridge [92])

In most cases a single agent has only partial control over its environment, which the agent exerts based upon its belief which action will satisfy the agent's design objectives best. Agents can therefore be said to be self-interested. There are trivial agents, such as control systems, as well as intelligent agents. According to Wooldridge [92, 93] intelligent agents can be characterised by fulfilling three prerequisites:

- *Reactivity* Intelligent agents have means of perception of the environment and can react to events or states within a time span that is adequate for the satisfaction of its design objectives.
- *Proactiveness* Intelligent agents do not restrict themselves to reacting to changes in the environment, but can take the initiative and act systematically towards a given goal (cf. Pitz [94]).
- *Social ability* Intelligent agents are able to communicate with other agents in the environment so as to cooperate or negotiate with these to satisfy their design objectives.

The first two prerequisites in the above list are suitable to enable an agent to act to reach a preferred state in the environment, either by performing immediate reactions or by means of systematic, proactive actions. The third prerequisite reflects an agent's necessity to cooperate and negotiate with other agents to reach it preferred state, or the best possible state considering a multitude of self-interested agents pursuing diverging objectives.

Decision Making in Multiagent Systems

In the literature, a number of negotiation protocols that enable agents to maximise relevant knowledge about the environment are described. Multiagent resource allocation protocols own certain properties dependent to their design. A list of desirable properties for negotiation protocols is proposed by Sandholm [95]:

- *Guaranteed success* A protocol guarantees success if it ensures to reach an agreement within a finite timespan.
- *Maximising combined Utility* Common utility is maximised if the protocol ensures the maximisation of the sum of utilities of all participating agents.
- *Pareto efficiency* A negotiation outcome is Pareto efficient (Pareto optimal) if no agent can increase its utility without decreasing at the utility of at least one other agent.
- *Individual rationality* A protocol is individual rational if every agent can improve its utility by adhering to the defined negotiation conventions.
- *Stability* A negotiation is strategically stable if no agent can improve by changing its negotiation strategy, i.e. from a cooperative to a non-cooperative strategy. A widely used stability definition is the Nash Equilibrium (cf. Nash [87], and Aubin [96])
- *Simplicity* As a general design primitive, simple protocols help to keep complexity low and make the decisions of individual agents comprehensible.
- *Distribution* Agents should negotiate directly, without the need for a central decision making instance, which would constitute a single point of failure.

The use of a decentralised multiagent resource allocation as opposed to a centralised decision making instance is discussed below.

2.3.5 Discussion of Decision Making

The literature on decision making shows two fundamental problems: First, all solutions on the Pareto frontier are optimal in an objective sense. This also applies as a converse argument, where all optimum solutions must necessarily be Pareto efficient. The problem is thus transformed towards the selection of only one optimum solution, which is inherently subjective. This then leads to the second problem, as a deliberation process under risk of injuring people cannot be made considering both ethical limitations and existing laws.

Still, a decision about a candidate region to be observed must be made in an active vision system. In this thesis the road more travelled is taken and our system is considered to be a driver assistance system as opposed to an autonomous driving vehicle, thus

providing additional information to the driver who in turn is still the final instance on all decisions of moral importance.

A number of functions to determine the combined utility from a set of objectives are presented. Whether these functions are used as such in a centralised decision making instance or the decision making is decentralised using agents must be decided. First, creating agents with singular interests is a means to divide a complex problem into a set of independent sub-problems, each of which can be solved robustly by considering a single problem domain. Sub-problems are said to be independent by Ephrati and Rosenschein [97] if the agents solving them do not need to interact to form their preference. Second using a decentralised group of agents making decisions avoids having a single point of failure, which has to be considered in safety-relevant applications. However the use of agents is more complex than using a predefined utility function. Therefore the use of a multi-agent resource allocation system is relegated to future work, which can be based upon our conclusions using a utility function in a central decision making instance.

2.4 Active Vision Systems

In sections 2.1 to 2.3, the integral parts of an active vision system are discussed. There, a basis for this section, which gives an overview over a variety of both biological and artificial active vision systems, is provided. Below a range of active vision systems is introduced, including the human visual system in section 2.4.1, saliency-driven vision systems in sections 2.4.2 to 2.4.4, and a utility-theoretical approach in section 2.4.5. A discussion of the reviewed active vision concepts is given in section 2.4.6.

2.4.1 Human Visual System

Many concepts from the human visual system (HVS), such as saccades, centre-surround detection, inhibition of return, or gaze-shifts driven by superior colliculus, are adapted in computer vision systems. In Tab. 2.4 the biological concepts used in the HVS are associated with the computer vision concepts described in sections 2.4.2 to 2.4.5. In the following, a brief overview of the HVS is given.

The HVS represents the part of the nervous system responsible for visual perception of the environment. For this, information from visible light is transformed and interpreted into a model of the visual environment. A schematic view of the HVS is given in Fig 2.16

Biological concept	Methods proposed by (*et al.*)
'Pop out' saliency	Treisman, Itti, Frintrop, Kadir
Saccadic gaze shifts	Trujillo, Itti, Frintrop, Koene
Centre-surround detection	Itti, Frintrop
Inhibition of return	Itti, Frintrop, Koene
Superior colliculus gaze shift	Koene

Table 2.4: Relevance of biological concepts for active vision systems proposed by Treisman [98], Itti *et al.* [99–101], Frintrop *et al.* [102, 103], Kadir *et al.* [104], Trujillo *et al.* [105], and Koene *et al.* [106].

and consists of (Schmidt *et al.* [107, 108]):

- two eyes, individually controllable by six exterior eye muscles
- a retina inside each eye, which itself consists of photoreceptor cells transforming visual light into neural signals
- two optic nerves, transmitting information from the retina to the brain
- the optic chiasma, a crossing of the optic nerves where information from both eyes is combined and split according to the visual field
- two lateral geniculate nuclei, where information from the optical nerves is projected to the primary visual cortex
- the visual cortex, which is reponsible for higher-level vision

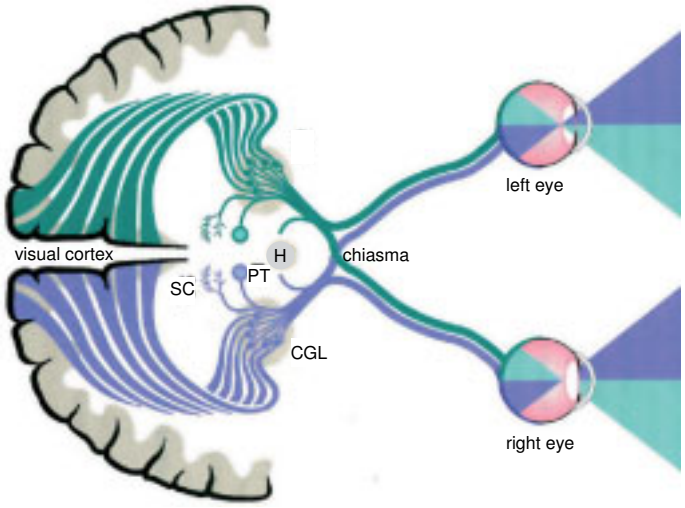


Figure 2.16: Schematic view of the optic tract of the human visual system. The optic nerves (green, blue) unite in the chiasma nervi optici. From there, the hypothalamus (H), the corpus geniculatum laterale (CGL), the area praetectalis (PT), the superior colliculus (SC), and the visual cortex are connected. Source: Schmidt and Lang [108].

Eyes and Gaze Shift

The HVS continuously observes the environment. The projection of the visual environment on the retina changes every 200 – 600 ms, due to eye, head, and body movements (cf. Schmidt and Lang [108]). These gaze shifts are coordinated by the prefrontal cortex (PFC, red region in Fig. 2.21), in two distinct modes; pre-conscious gaze changes and conscious gaze changes (cf. Henderson [109], and Einhäuser *et al.* [110]). Pre-conscious gaze changes are performed if the nature and geometry of an object is still unknown. After the HVS has had enough time to localise and classify objects in the environment, the gaze direction is consciously directed towards the current object of interest.

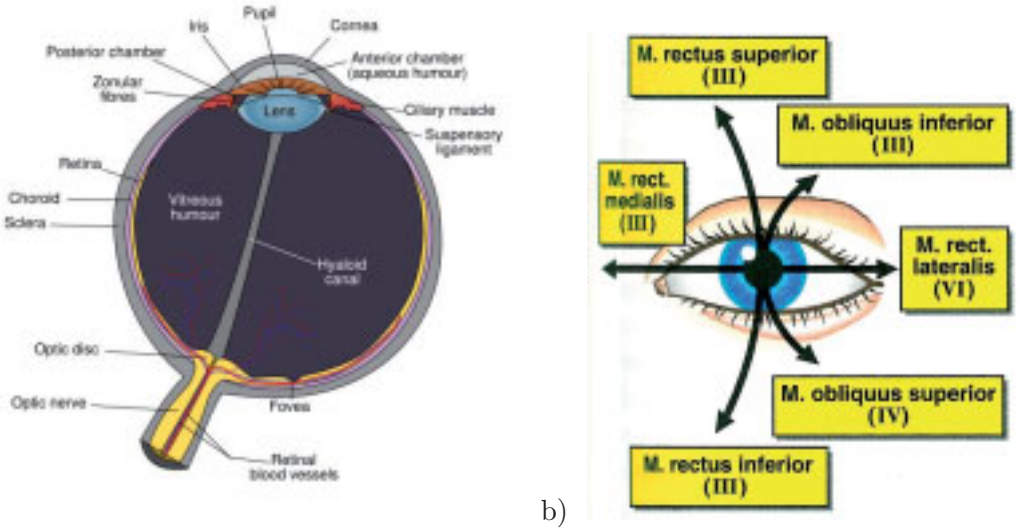
Nixon and Aguado [111] point out that the receptors' density on the retina is non-uniform. Eye movements direct the area of highest visual acuity towards the object of highest interest for the current moment. The part of the retina specialised for accurate vision (fovea, cf. Fig. 2.17a) contains a 4 – 40 times higher receptor density and covers a field of vision of around 5°, whereas the maximum receptor density ($1.6{\cdot}10^5$ cells/mm^2) is located within 1° of the fovea's centre (fovea centralis, cf. Schmidt and Lang [108]). If the object's spatial expansion exceeds the fovea's field of vision, it is observed sequentially using small, jerky gaze shifts named *saccades*.

Human eye movements such as saccades are conducted by six exterior eye muscles horizontally (musculus rectus medialis and m. rectus lateralis) as well as vertically and rotationally (m. rectus superior, m. rectus inferior, m. obliquus superior, and m. obliquus inferior, cf. Fig. 2.17b).

Eye movements can be grouped into three classes with different temporal dynamics: saccades, fixation and smooth pursuit.

Saccade During exploration of the environment, the human eye moves in rapid, jerky movements from one fixation point to the next every 10 – 80 ms. The amplitude of these saccades can be as little as 0.03° to 2° (microsaccades), but can reach shift angles of 90° and more. The mean angular velocity of the eye depends upon the saccade's amplitude and exceeds 500°/s for large saccades (>60°, cf. Schmidt and Lang [108]). Tracked saccades of the HVS are also given in Fig. 2.38.

Fixation In between saccades, periods of fixation of 200 – 600 ms occur. The observation of the fixated object is performed during these periods.



a) b)

Figure 2.17: Fig. a) shows the schematic diagram of the human eye. Fig. b) shows the interaction of the six exterior human eye muscles responsible for eye movements. Source: Schmidt and Scheible [107].

Smooth Pursuit A moving object can be pursued by the eye with a smooth pursuit motion. By this, the observed object remains in the fovea centralis region. The angular velocity of the smooth pursuit is approximately that of the pursued object if the latter is not faster than 100°/s. For higher angular velocities, correcting saccades and head motions assist the pursuit motion.

Processing in the Human Visual System

The function of the HVS can be represented using an information pyramid. Throughout processing, the amount of visual data is considerably reduced. From initially 10^{10} bits/s on the retina, only 10^4 bits/s of information reach attentive scrunity (cf. Fig. 2.18, Anderson *et al.* [6]).

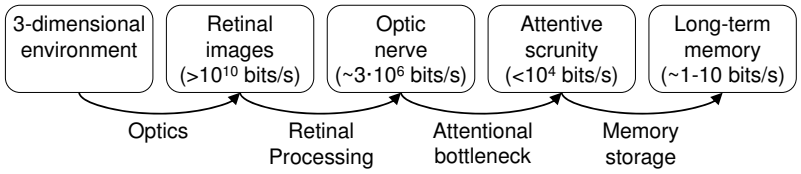


Figure 2.18: Information pyramid for the visual system as given in Anderson *et al.* [6], therein based upon [112–115].

In the retina, different types of photoreceptor cells are specialised for different visual

tasks. Magnocellular photoreceptors are sensitive towards spatial contrast and motion but cannot distinguish colours. The smaller parvocellular phororeceptors are sensitive towards colour, and exhibit a much higher visual acuity. Photoreceptor cells constitute the first of five layers of processing in the retina (cf. Fig. 2.19, Masland [116]).

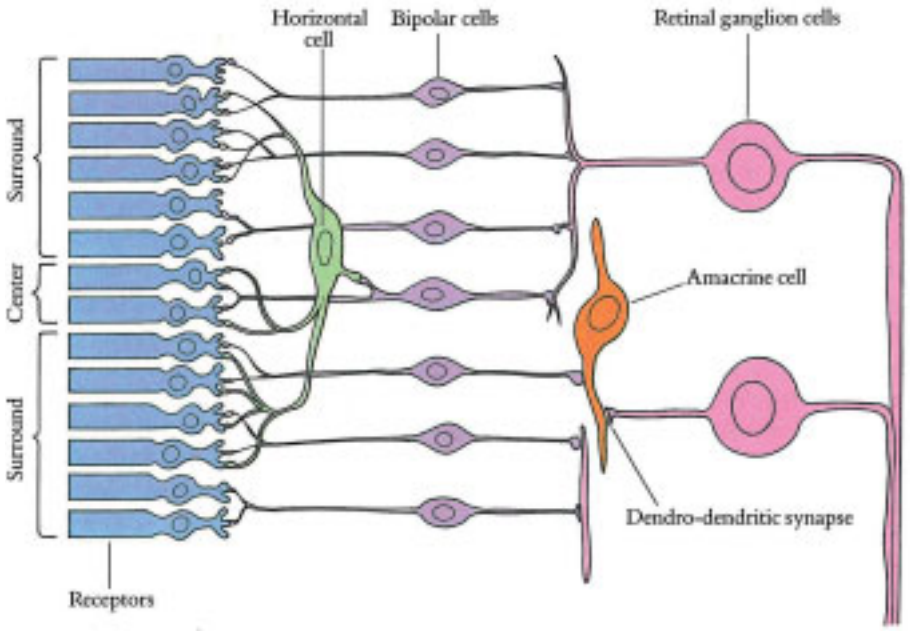


Figure 2.19: Layout of the visual signal pathway over five retinal layers, organised in a centre region and a surround region of photoreceptors. Note that the light is coming from the right hand side and has to pass through all four cell layers before reaching the photoreceptors. Source: Hubel [117].

The photoreceptors' horizontal cells (second layer) provide inhibitory input to bipolar cells (third layer) and combine information from surrounding photoreceptor cells (each region is compared to its surrounding region's red-greenness, blue-yellowness, and black-whiteness, see Fig. 2.20). In the third layer, bipolar cells receive inhibitory input from the horizontal cells as well as excitatory input from photoreceptor cells. The function of the amacrine cells in the fourth layer is a field of ongoing investigation. According to Masland [116] the function of amacrine cells is to contribute to inhibitory surrounds to both bipolar cell and ganglion cell layers. Finally the ganglion cell layer transmits visual information to the optic nerve by firing action potentials.

The information transmitted over the optic nerves unites in the chiasma nervi optici (see Fig. 2.16). The left visual hemisphere is transmitted towards the brain's right hemisphere and the right visual hemisphere transmitted towards the brain's left hemisphere.

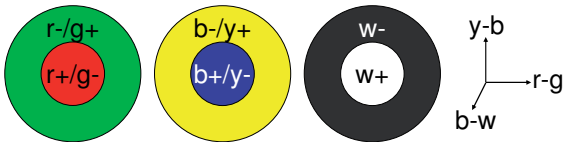


Figure 2.20: Centre surround fields used by the human visual system (converse fields not shown). Each region is compared to its surrounding region's red-greenness, blue-yellowness, and black-whiteness. In dim light, when colour information is no longer available, this allows observation of luminance using only the *b*-*w* axis, as opposed to a red-green-blue (RGB) colour system. Source: Hubel [117].

The optic nerve is connected to the corpus geniculatum laterale (CGL), which transmits information about shape, colour, range, and motion towards the visual cortex. The CGL also transmits into the superior colliculus (SC), where, bypassing the visual cortex, reflexive eye saccades are initiated by transmitting directly to the prefrontal cortex (PFC, red area in Fig. 2.21) responsible for coordinated eye-movements.

The visual cortex (Fig. 2.16 and orange areas in Fig. 2.21) is divided into a primary visual cortex (V1) and three extrastriate visual cortical areas (V2 to V4).

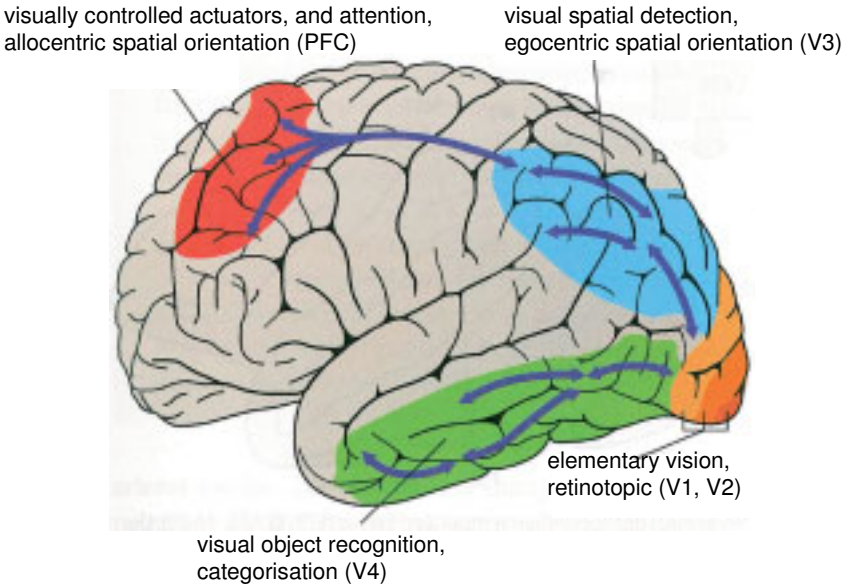


Figure 2.21: Two streams of visual cognitive functions originating from the visual cortex (V1 and V2, orange): the dorsal stream is associated with motion, spatial orientation (V3, blue), and control of actuators (PFC, red). The ventral stream (V4, green) is associated with object recognition and categorisation. Source: Schmidt and Lang [108].

Primary Visual Cortex (V1) Visual area V1 (Fig. 2.21, orange areas) is organised retinotopically. Retinotopy ensures that neighbouring areas on the retina are mapped

towards neighbouring neurons in area V1. Given this property, orientation, direction, and colour can be determined by the interaction of neighbouring neurons. As a direct mapping of retinal cells, a high density of photoreceptor cells inside the fovea results in an accordingly large number of neurons in the area V1 mapping to the fovea.

Prestriate Cortex (V2) Information from V1 is processed in V2 (Fig. 2.21, orange areas) in specific subsystems that analyse colour, shape, motion and range of static patterns. The visual processing then splits into two pathways; the ventral stream V3, and the dorsal stream V4.

Dorsal Stream (V3 & PFC) The neurons in V3 (Fig. 2.21, blue area) are specialised in determining the motion and range of object contours provided by V1 and V2. The dorsal stream also communicates with the prefrontal cortex (PFC, Fig. 2.21, red area), where visually controlled actuators and attentional functions are located.

Ventral Stream (V4) Colour specific neurons in V1 and V2 transmit into V4 (Fig. 2.21, green area) and the adjacent inferior temporal lobe. There, objects are recognised by their characteristic colours and colour contrasts.

2.4.2 Bottom-Up Saliency Driven Vision Systems

Active vision systems aiming to imitate the pre-conscious gaze control mechanism of the human eye frequently use a bottom-up saliency method. The term bottom-up refers to an untrained saliency operator without any explicit prior knowledge. Below a definition of saliency is given and six bottom-up saliency detectors are presented.

Definitions of Saliency

In the literature, saliency is often derived from the fixation patterns of the human eye which, during its pre-attentive phase, treats regions as salient, which 'pop out' (cf. Treisman [98]) of their surroundings. The saliency operators proposed by Itti *et al.* [99, 100], Frintrop *et al.* [102, 103], and Kadir *et al.* [104] use this 'pop out' criterion. This definition follows the idea of evaluating the local contrast between a region and its surrounding regions.

A different definition treats regions as salient, whose feature space representation is rare, at best unique, in their environment. This form of saliency is used by Walker *et al.* [118], and Collomosse and Hall [119]. The latter definition assumes statistical knowledge and determines saliency in a global context.

The difference between saliency emerging from a global rarity and local contrast is illustrated in Fig. 2.22.

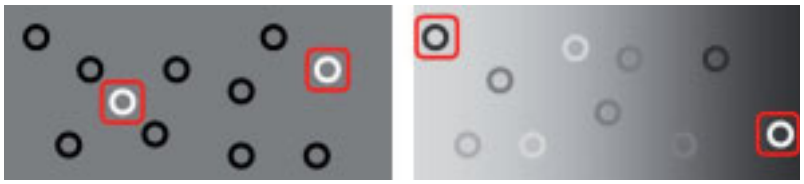


Figure 2.22: Saliency can emerge from both global rarity (left) and local contrast (right).

The notion of *surprise* as a saliency measure proposed by Baldi [120, 121] is unconventional and is discussed in more detail later.

Centre-Surround Saliency Approach by Itti *et al.*

The method presented by Itti *et al.* [99, 100] uses a local centre-surround approach inspired by the neuronal architecture of the early primate visual system. This approach requires four sequential processing steps

1. conversion into feature space,
2. centre-surround receptive field profiles,
3. combining information across multiple maps, and
4. fusion of conspicuity maps,

which can be seen in Fig. 2.23 and are discussed in more detail below.

Conversion into Feature Space Initially the input image is converted to feature space using linear operators to specific stimulus dimensions, such as luminance, colour, or local orientations at decreasing scales of the input image.

As described by Burt and Adelson [122], a Gaussian pyramid for different spatial scales is created by progressively down-scaling the input images. Nine scales are implemented ranging from 1:1 at level 0 to 1:256 at level 8. Down-scaling of the input images is

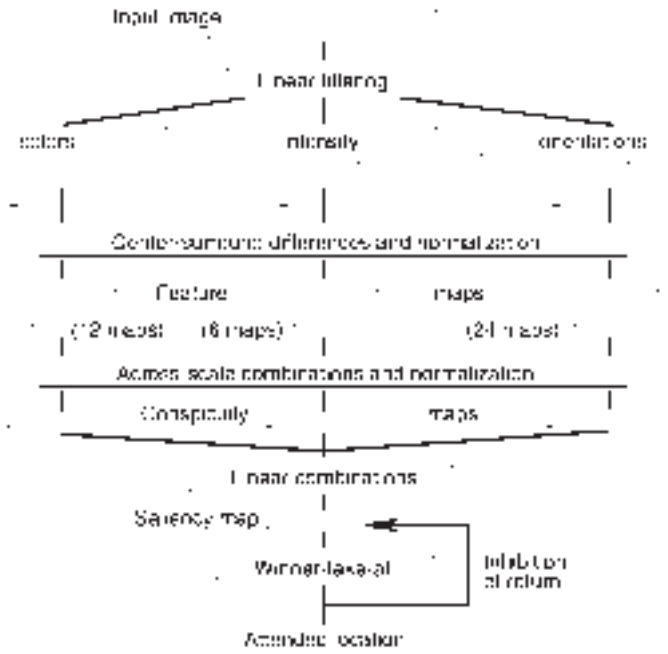


Figure 2.23: General architecture of the centre-surround saliency approach proposed by Itti *et al.* [99]. Source: Itti [101].

performed using a 5×5 Gaussian filter on the source image and then sub-sampling it by a factor of 2.

As a first feature, luminance is computed at all scales using a simple addition of the red, green, and blue colour channels. For hue, the luminance image is normalised and four broadly tuned colour channels are created; red, green, blue, and yellow. Each channel responds maximally to its specific colour and with zero for black or white inputs. Local orientation is obtained using Gabor pyramids [123] at four preferred directions: 0°, 45°, 90°, and 135°.

Centre-Surround Receptive Field Profiles A centre-surround operation, inspired by the visual receptive fields in the HVS (cf. Fig. 2.20), is performed by calculating the differences between maps of the same feature at different scales. For this, the coarse map is interpolated towards the finer scale and then subtracted. Differences are calculated for six scale pairs ranging from scale 2 to scale 8 and a scale interval of 3 and 4 scales (i.e. 2-5, 2-6, 3-6, 3-7, 4-7, 4-8). Centre-surround feature maps are determined for seven types of features also used in the HVS: on/off image intensity contrast (e.g. Leventhal [124]), red/green and blue/yellow double opponent channels (e.g. Hubel [117], and Engel *et al.* [125]), and 4 local orientation contrasts (e.g. DeValois *et al.* [126], and Tootell *et al.* [127]).

The result of the centre-surround operator are a total of 42 maps: 6 intensity maps, 12 colour maps, and 24 orientation maps.

Combining Information Across Multiple Maps In Itti *et al.* [99] a contents-based global nonlinear amplification is proposed. Each map is normalised separately to a range $[0 \ldots M]$. The feature map is then mutliplied by $(M - \overline{m})^2$, where m is the mean value of all local maxima inside the feature map. This step introduces the notion of global rarity by promoting maps with large differences between M and $\overline{m}$ (cf. Fig. 2.24).

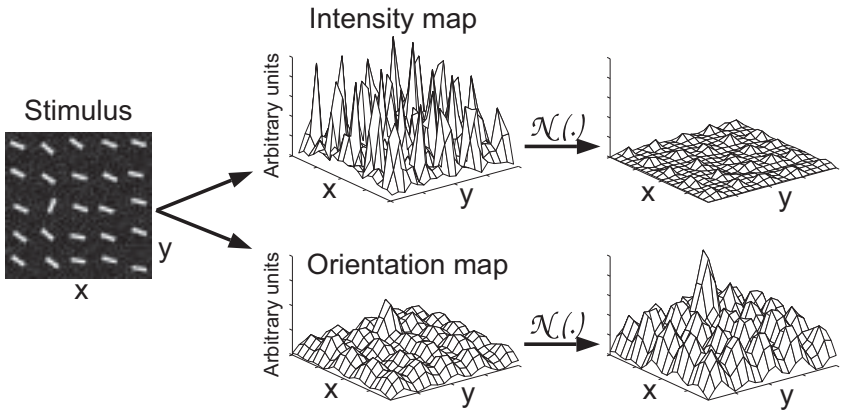


Figure 2.24: Contents-based global nonlinear amplification proposed by Itti *et al.* [99]. Source: Itti [101].

Apart from the global nonlinear amplification method, an iterative competition scheme is proposed by Itti [101]. There, the inhibitory effect of neighbouring neurons is modelled using a 2-D Difference-of-Gaussians (DoG) pattern. Iterative convolutions of each map with a DoG pattern causes local maxima to be attenuated by neighbouring local maxima if these exhibit similar values. If the greater neighbouring area has much lower values than the convolution centre, attenuation is only marginal. At the same time, fields that are very close to the convolution centre reinforce the local maximum. These two effects cause local maxima of average amplitudes to diminish and local maxima with amplitudes above average to remain or even rise (cf. Fig. 2.25).

Fusion of Conspicuity Maps The normalised maps are combined across scales into three separate conspicuity maps for intensity, colour, and orientation. This is performed with a simple addition at scale 4. The use of three maps is motivated by Itti [101] with the hypothesis that

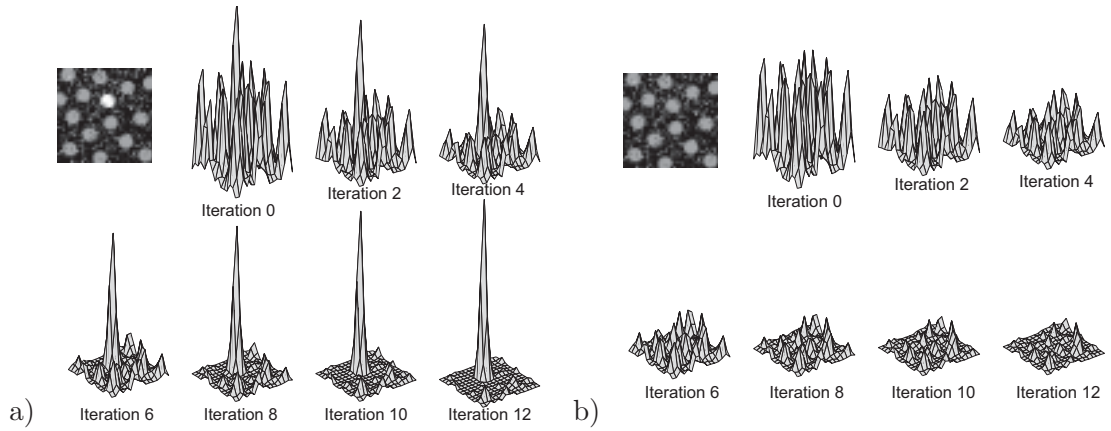


Figure 2.25: Iterative competition scheme in Itti [101] for a) one distinct global maximum and b) many similar local maxima. Source: Itti [101].

> *"... similar features compete strongly for saliency, while different modalities contribute independently to the saliency map."* (Itti [101])

The resulting combined map is then used as saliency map.

Centre-Surround Saliency Approach by Frintrop *et al.*

The method used in Frintrop *et al.* [102] is similar to Itti *et al.* [99], with the exception of the map weighting concept illustrated in Fig. 2.26. Instead of a difference $(M - \overline{m})^2$, the square root of the number of local maxima $\sqrt{N_{max}}$ is used as a normaliser by which the map is divided. This concept promotes maps with few local maxima, regardless of the actual values of the maxima.

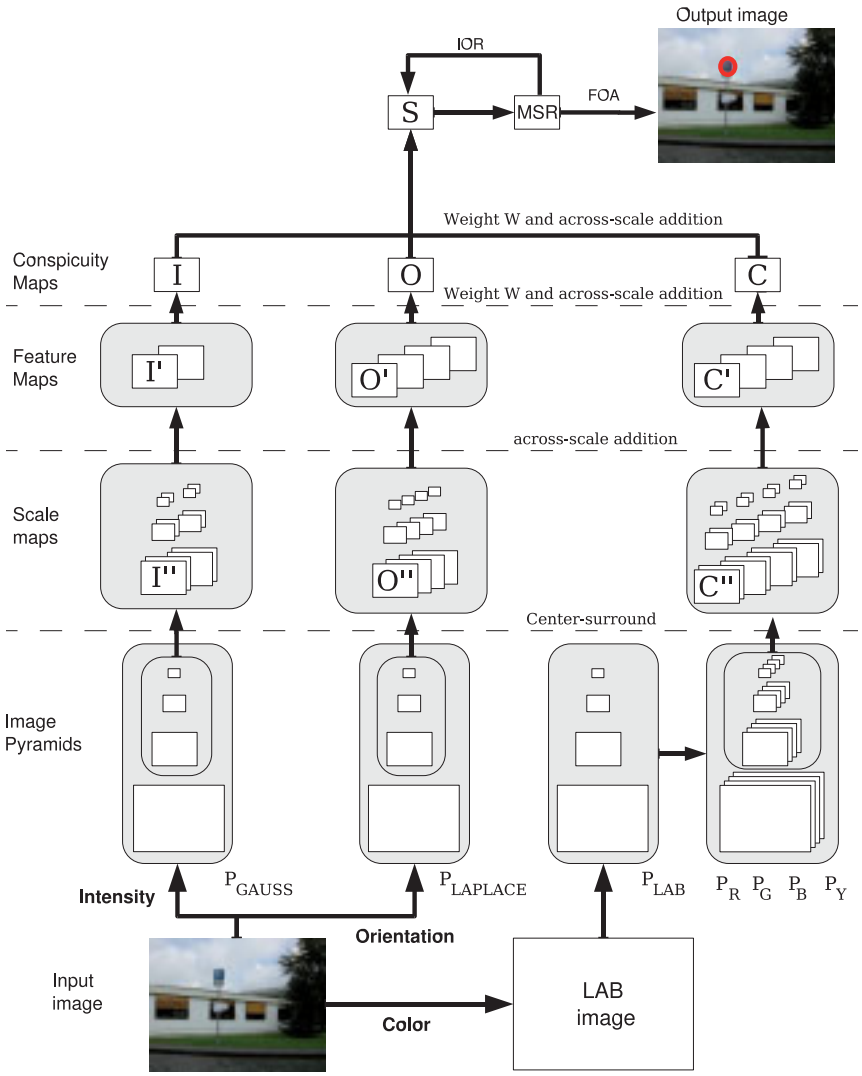


Figure 2.26: Schematic concept of the model proposed by Frintrop *et al.* [102] detecting a single region of interest marked red in the output image. Source: Frintrop [103].

Superior Colliculus Gaze Shift Method by Koene *et al.*

A multi-modal gaze shift model inspired by the superior colliculus (SC, cf. Fig. 2.16) in the HVS is proposed by Koene *et al.* [106]. Electrophysiological and behavioural studies on primates by Arai and Keller [128] show that a weighted summation of the excitatory multi-modal (eyes and ears) sensory and voluntary inputs can be used as a model for SC. An inhibitory input sets an activation threshold, which must be exceeded to influence the gaze shift. This inhibitory component is introduced in the model by subtraction from the individual inputs, with all negative values set to zero.

The resulting weighted summation of the individual gaze shift vectors in the SC module

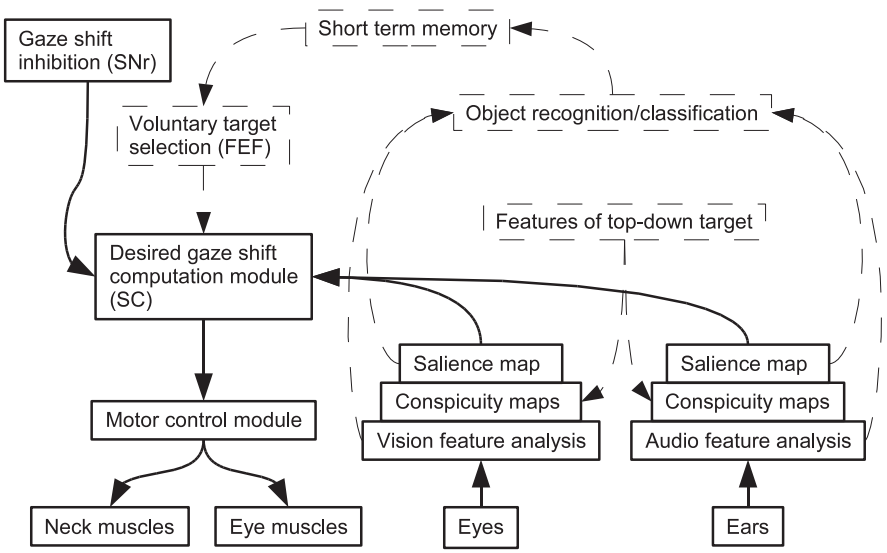


Figure 2.27: Schematic overview of the multi-modal superior colliculus gaze shift method. Source: Koene *et al.* [106].

presents the desired gaze shift vector relative to the actual gaze direction. This is in contrast to Itti *et al.* [99], Frintrop *et al.* [102], and other stimulus driven active vision systems (e.g. Lee *et al.* [129], Koch and Ullman [130], and Li [131]), which use a winner-take-all process in choosing a gaze shift direction rather than combining individual gaze shift proposals.

Winner-take-all selection always chooses the most salient selection as target for the next gaze shift. This results in certain disadvantages that are pointed out by Koene *et al.* [106]:

- In a neural system, winner-take-all is usually implemented as an iterative algorithm, which is problematic if rapid responses are required.
- Winner-take-all does not reflect the relative saliency of the chosen gaze direction in comparison to other gaze directions.
- In binocular vision systems it is necessary to average the information from the left and right eye before a winner-take-all algorithm can be applied.

A drawback of the SC gaze shift model is that if multiple highly salient stimuli are simultaneously present, the centre of weight lies between the salient locations. This is an effect also observed in primate saccades under the same conditions by Arai and Keller [128]. Koene *et al.* [106] observe that this event is rare, due to a high inhibitory input and a resulting low probability of events surpassing the inhibitory threshold.

Affine invariant Saliency by Kadir *et al.*

A saliency algorithm that operates across feature-space and scale-space is presented by Kadir and Brady [132, 133]. Its underlying principle is, that salient regions are considered to exhibit a high entropy both in their local attributes and over spatial scale. This concept is illustrated in Fig. 2.28.

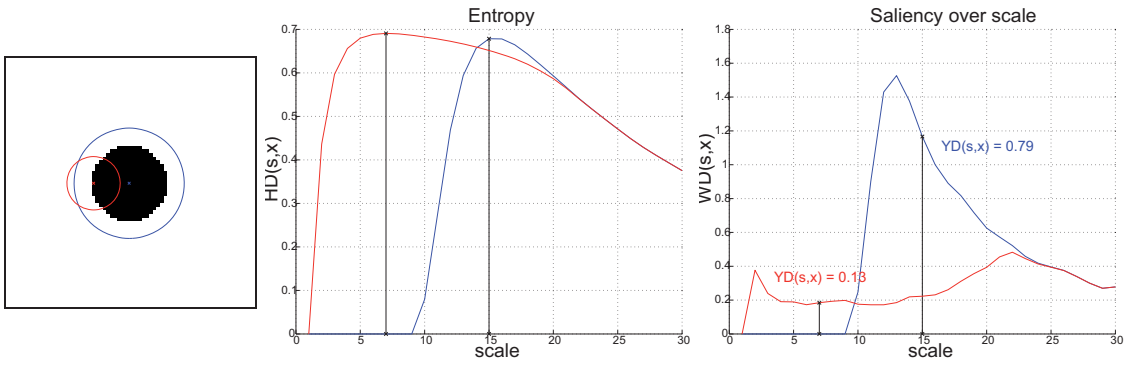


Figure 2.28: Example of affine invariant saliency by Kadir *et al.* [104]. Entropy $\mathcal{H}$ and saliency over scale $\mathcal{Y}$ corresponding to the circle's centre (blue) and the circle's edge (red) in the source image are given. Source: Kadir *et al.* [104].

In Fig. 2.28 two entropy $\mathcal{H}$ peaks corresponding to the circle's centre and its edge across scale ς are found. The saliency over scale $\mathcal{Y}$ graph is the product of an inter-scale unpredictability $\mathcal{W}$ and the entropy $\mathcal{H}$ using Eq. 2.17.

$$\mathcal{Y}(i,j,\varsigma) = \mathcal{W}(i,j,\varsigma) \cdot \mathcal{H}(i,j,\varsigma) \tag{2.17}$$

The advantage of this concept is that saliency is not only determined for a pixel (i,j) but also contains information about the salient region's scale (i,j,ς). The saliency detector presented by Kadir and Brady [132, 133] is extended towards an affine invariant saliency detector by Kadir *et al.* [104]. The latter method is also described and compared with other affine region detectors by Mikolajczyk *et al.* [134, 135].

Statistical Rarity as a Saliency Measure

Statistical rarity as a saliency measure has not been explicitly applied for an active vision concept but is introduced below as it presents a relevant approach towards detecting salient regions in the environment.

Walker *et al.* [118] compute the Mahalanobis distance d between a local feature vector

x and the environment's mean feature vector $\bar{x}$ using

$$d(x, \bar{x}) = \sqrt{(x - \bar{x})^T S^{-1} (x - \bar{x})} \tag{2.18}$$

where S^{-1} is the inverse covariance matrix of all feature vectors x.

The Mahalanobis distance between x and $\bar{x}$ is large if a feature vector diverges from the mean feature vector. While Walker *et al.* [118] use the Mahalanobis distance itself as a saliency measure, Collomosse and Hall [119] propose using the squared Mahalanobis distance d^2 as saliency.

Regions regarded as salient using this measure are regions with feature combinations that are rare, at best unique, in the environment. A high local contrast is therefore only considered salient if there is little local contrast in the environment.

Surprise Saliency by Baldi *et al.*

A saliency method utilising the notion of surprise is presented by Baldi [120, 121]. There it is argued that shifting attention is a rapid process that is likely to be driven by bottom-up cues rather than top-down cues. The concept of surprise as a saliency measure examines the difference between a prior and a posterior probability distribution.

As an example, a single probability for a traffic participant $\mathcal{TP}$ changes from a prior probability $P_{k-1}(\mathcal{TP})$ to a posterior probability $P_k(\mathcal{TP}|C)$ dependent upon the outcome of classifier cascade $\mathcal{C}$ (cf. section 6.1.2 where our proposed system performs this calculation for all traffic participant types).

$$P_k(\mathcal{TP}|C) = \frac{P(C|\mathcal{TP}) P_{k-1}(\mathcal{TP})}{P(C)} \tag{2.19}$$

Surprise $S(C, \mathcal{TP})$ is then defined by Baldi [120] as the log odd ratio

$$S(C, \mathcal{TP}) = \log \frac{P_{k-1}(\mathcal{TP})}{P_k(\mathcal{TP}|C)} \tag{2.20}$$

Surprise saliency is somewhat unconventional as bottom-up saliency is gained by examining semantic information such as detected objects. However it is still considered a bottom-up system in this context since the calculation of surprise saliency $S(C, \mathcal{TP})$ itself is independent of prior knowledge. In contrast to the statistical rarity approach by Walker *et al.* [118] or Collomosse and Hall [119], surprise examines the probabilistic difference over

time rather than spatial statistical rarity (see Fig. 2.29).

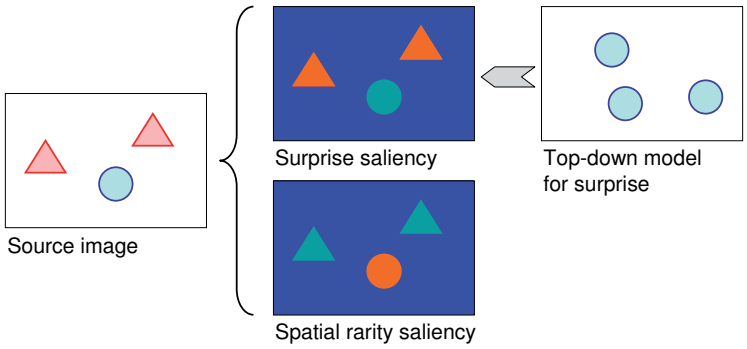


Figure 2.29: Schematic comparison of surprise saliency and spatial rarity saliency. Surprise saliency considers red triangles as more salient since the top-down model only contains blue circles. If the spatial rarity is considered, a single blue circle is more salient than two red triangles.

2.4.3 Top-Down Saliency Driven Vision Systems

A top-down approach to saliency implies a predefined set of objects to be regarded as salient. This implication requires prior knowledge about observable objects' properties that can either be provided manually (e.g. by a set of rules), or trained using a machine learning algorithm. In the following, two dedicated top-down saliency systems and two comparable object recognition systems are presented.

Top-down Saliency by Navalpakkam and Itti

Navalpakkam and Itti [136] propose a top-down saliency measure that maximises the signal-to-noise ratio between a search target and distractors. This approach requires knowledge about bottom-up saliency to optimise the signal-to-noise ratio. Methods incorporating both bottom-up and top-down saliency are discussed in section 2.4.4, where the optimal cue selection strategy presented by Navalpakkam and Itti [136] is extended towards an integrated model in Navalpakkam and Itti [137].

The top-down mechanism discussed by Navalpakkam and Itti [136] aims at the determination and use of a top-down factor g. As an intuitive result, Navalpakkam and Itti [136] state that g_i increases as $\frac{SNR_i}{SNR}$ increases, where SNR_i represents the signal-to-noise ratio of the i^{th} bottom-up saliency map. From the simplification that g_i is considered proportional towards $\frac{SNR_i}{SNR}$ follows

$$g_i \propto \frac{SNR_i}{SNR} \tag{2.21}$$

therefore the top-down factor g_i is optimised by a maximisation of the signal-to-noise relation. This top down information is used by Navalpakkam and Itti [137] which is presented later in section 2.4.4.

Top-down Saliency by Frintrop *et al.*

In Frintrop *et al.* [103, 138] a top-down saliency detector is trained by learning target-relevant features as well as background features from a single training image. For goal-directed search, target-relevant features are used to determine an excitatory map, whereas background features are used to calculate an inhibitory map. Top-down saliency is then calculated by subtracting the inhibitory map from the excitatory map, saturating all negative results to zero. An example for top-down saliency is given in Fig. 2.30.

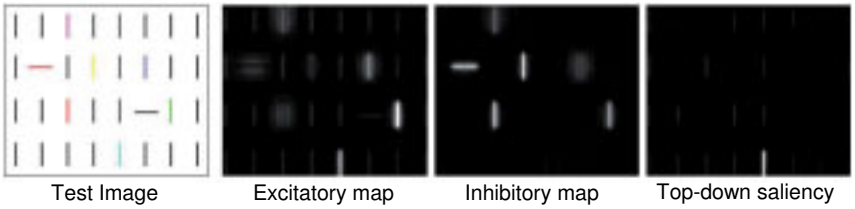


Figure 2.30: Top-down saliency calculation of the search for a vertical cyan bar in the test image. The excitatory map shows the presence of target-relevant features, while the inhibitory map considers all green bars as background. Subtracting the inhibitory map from the excitatory map, a top-down map is calculated. Source: Frintrop *et al.* [103, 138].

This top-down saliency measure uses the feature maps also used for bottom-up saliency. While for the bottom-up saliency a uniqueness weight $(N_{max})^{-\frac{1}{2}}$ considering the number of local maxima N_{max} is used, trained feature weights are used for top-down saliency.

First, the region of interest is manually labelled and bottom-up saliency is computed as proposed by Frintrop *et al.* [102]. Second, the most salient region (MSR) in the region of interest is determined using bottom-up saliency information. Third, for every feature map and conspicuity map X_n, the mean value inside $m_{n(MSR)}$ and outside $m_{n(\neg MSR)}$ the MSR is calculated and the weight w_n is determined by a division of both mean values:

$$w_n = \frac{m_{n(MSR)}}{m_{n(\neg MSR)}} \tag{2.22}$$

In Frintrop *et al.* [103, 138] weights of $w_n > 1$ will contribute to the excitatory map (E)

and weights of $w_n < 1$ will contribute to the inhibitory map (I). Features with a weight of $w_n = 1$ are disregarded, as these cannot be used to distinguish between object and background.

$$E = \sum_{n:w_n>1} (X_n \cdot w_n) \tag{2.23}$$

$$I = \sum_{n:w_n<1} (\frac{X_n}{w_n}) \tag{2.24}$$

Top-down saliency S_{td} is then calculated by subtracting the inhibitory map from the excitatory map using Eq. 2.25.

$$S_{td} = E - I \tag{2.25}$$

Trained Cascades by Viola and Jones

A trained classifier cascade such as the Viola and Jones face detector [52, 139] can also be considered a top-down saliency algorithm. Cascaded classifiers disregard a large portion of negative samples, and thus non-salient regions in a top-down definition, at every stage.

The algorithm's computational efficiency pointed out in section 2.2.1, its shown robustness, and the desirable 'side-product' of obtaining a list of detected objects suggest the use of a trained classifier cascades in applications where a list of detected object positions besides top-down saliency information is considered an advantage.

Focused Vision Based Approach by Trujillo *et al.*

The focused vision based approach by Trujillo *et al.* [105] emulates the active process of the human eye by recognising objects in a saccadic object part recognition pattern influenced by previously detected object parts. This influence is exerted by means of a covariance matrix with learned statistical relationships between object parts. The example given in Trujillo *et al.* [105] describes the guided recognition of a face, where both colour and vertical position of the eye found first are highly correlated to the second eye, effectively constraining the search space for the second eye.

An interesting property of the focused vision based approach is its highly focused exploration of the source image together with a strong emphasis on learned saccadic gaze

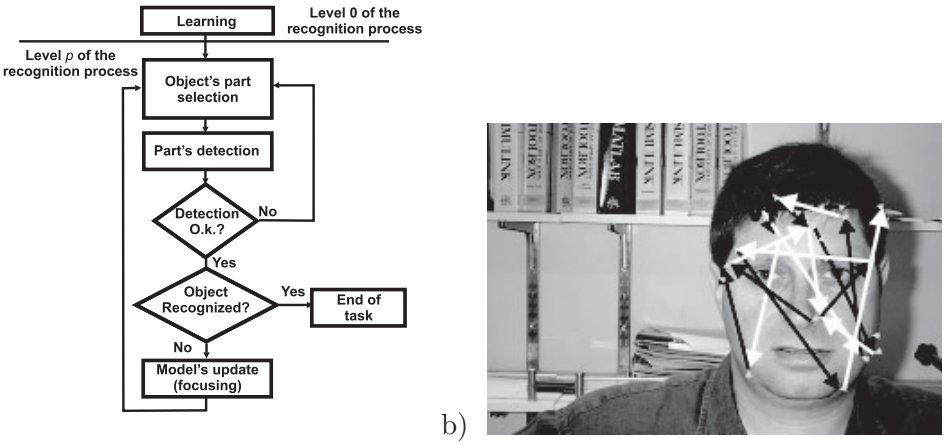


Figure 2.31: a) Flow diagram of the focused vision based approach by Trujillo *et al.* [105]. Figure b) shows a sequence of saccadic shifts during recognition of a face. Source: Trujillo *et al.* [105].

shifts. Of all reviewed active vision methods the focused vision based approach emulates the HSV's saccadic exploration best. Apart from faces, the presented approach is also used for vehicle classification in Trujillo *et al.* [140].

2.4.4 Combined Bottom-Up and Top-Down Vision Systems

A number of combined bottom-up and top-down systems can be found in the literature. There, top-down and bottom-up information is combined into an overall saliency map. In the following, three hybrid systems are presented.

Integrated Model by Navalpakkam and Itti

An approach to integrate top-down and bottom-up attention is proposed by Navalpakkam and Itti [137]. The model combines both top-down cues (cf. Navalpakkam and Itti [136]) and bottom-up cues (cf. Itti *et al.* [99]) to guide visual attention while searching for a target in a cluttered environment.

The combination of cues is performed using a linear combination of bottom-up saliency for individual features modulated by a top-down gain factor by multiplicative gain modulation (e.g. Treue and Martinez-Trujillo [141, 142]) and integrated across all dimensions as proposed in the guided search model by Wolfe [143]. The combined saliency S is determined using

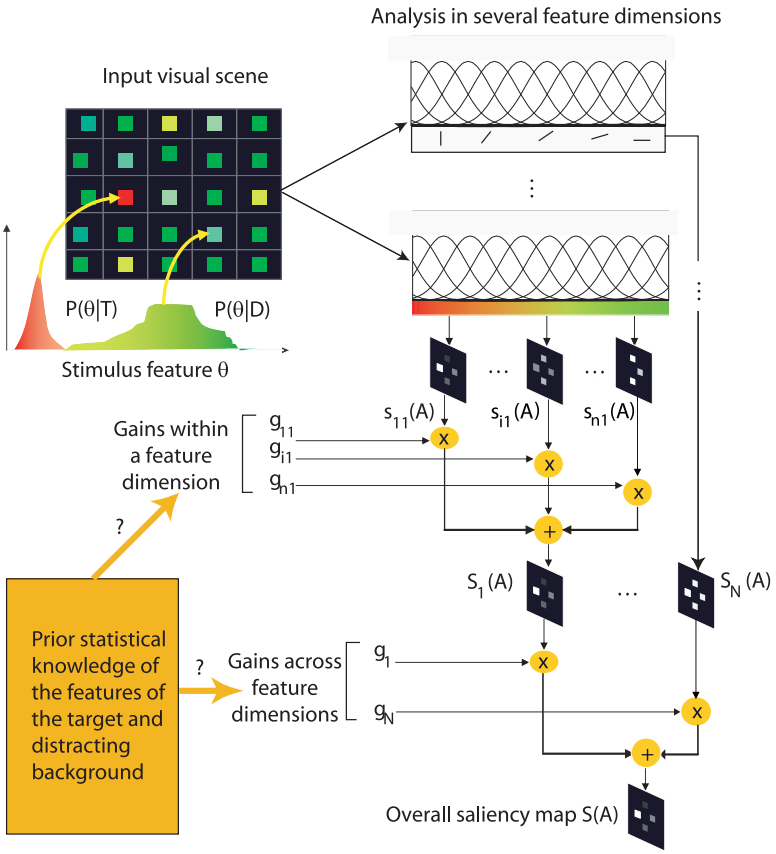


Figure 2.32: Integrated model method by Navalpakkam and Itti [137]. Source: Navalpakkam and Itti [137].

$$S(i,j) = \sum_{a=1}^{N} g_a \sum_{a=1}^{n} g_{ba} S_{ba}(i,j) \tag{2.26}$$

where S_{ba} is the bottom-up saliency and g_a the top-down factor for n saliency maps over N feature dimensions. In Fig. 2.32 a schematic overview of the integrated model method is shown.

Goal-directed search by Frintrop

In Frintrop *et al.* [25, 103, 144] a visual attention system for object detection and goal-directed search (VOCUS) is presented. It combines a bottom-up saliency method (Frintrop *et al.* [102]) with a top-down saliency method described by Frintrop *et al.* [138]. An overview of the VOCUS method can be seen in Fig. 2.33.

Combination of bottom-up saliency S_{bu} and top-down saliency S_{td} is performed using a weighted sum of both maps, following the guided search model by Wolfe [143]. After

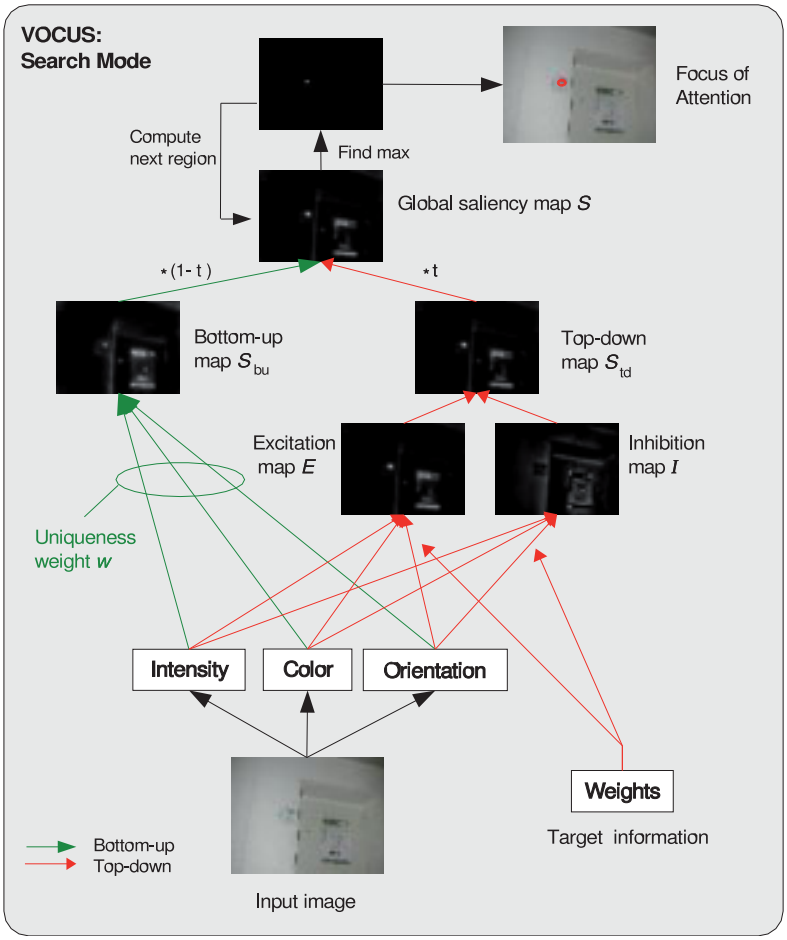


Figure 2.33: Goal directed search method (VOCUS) by Frintrop [103]. Source: Frintrop [103].

normalisation of both maps towards the same range, the saliency maps are fused using a top-down factor $t \in [0, 1]$.

$$S = (1 - t) \cdot S_{bu} + t \cdot S_{td} \tag{2.27}$$

The resulting global saliency map S is used to determine the most salient region and focusing on this region employing a winner-takes-all strategy. The optimal value and use of a top-down factor t is considered problematic by Frintrop [103], as the fusion of bottom-up and top-down information in human perception is considered to be unclear. This motivates the use of a 'concentration factor' t, indicating the influence of bottom-up induced attentional capture on the global saliency map as described by Theeuwes [145].

Contextual Guidance Model by Torralba *et al.*

Torralba *et al.* [146] describe a contextual guidance model that combines both global gist analysis by performing a principal component analysis (PCA) on the whole image (cf. Oliva and Torralba [147]) and local saliency using the algorithm proposed by Itti *et al.* [99] at an early stage of visual processing. An overview of the contextual guidance model is given in Fig. 2.34.

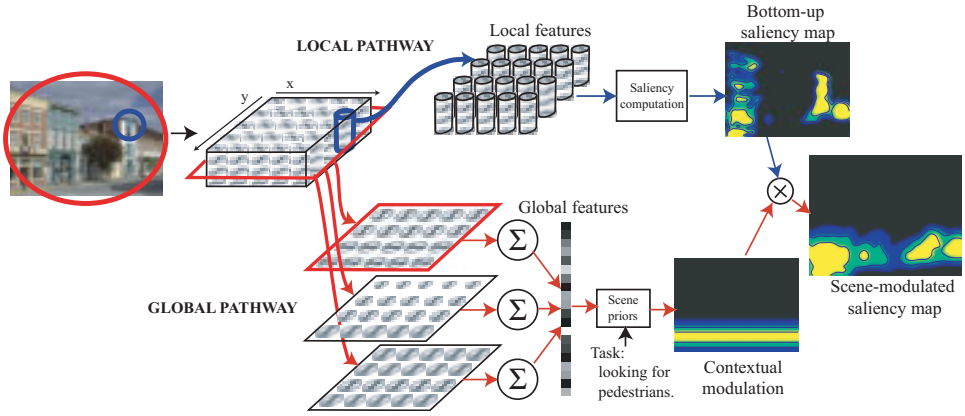


Figure 2.34: Contextual Guidance model proposed by Torralba *et al.* [146]. Source: Torralba *et al.* [146].

While the generation of the bottom-up saliency map in Fig. 2.34 using a centre-surround operator has been discussed above, gist analysis using PCA is described by Oliva and Torralba [147, 148]. There it is argued, that the gist of complex scenes can be determined from coarse spatial representation of the entire image without prior scene segmentation and object detection.

The detected gist is then used to determine a horizontal region in the image, where objects of a given class are most likely to appear, following the notion that an ideal observer will search the most likely positions in the image first, which is used as a top-down cue (cf. contextual modulation in Fig. 2.34).

Bottom-up and top-down probabilities are then combined using a weighted multiplication

$$S = (S_{bu})^{-\gamma} \cdot S_{td} \tag{2.28}$$

where γ is a trained parameter that is determined to be optimal in the range $[0.01, 0.3]$ by Torralba *et al.* [146].

A performance comparison on a detection task between bottom-up saliency, top-down context alone, and the full contextual guidance model by Torralba *et al.* [146] is given in Fig. 2.35. There it can be seen that the performance of the contextual knowledge, while performing well, does not show a statistically significant difference towards the use of top-down context alone.

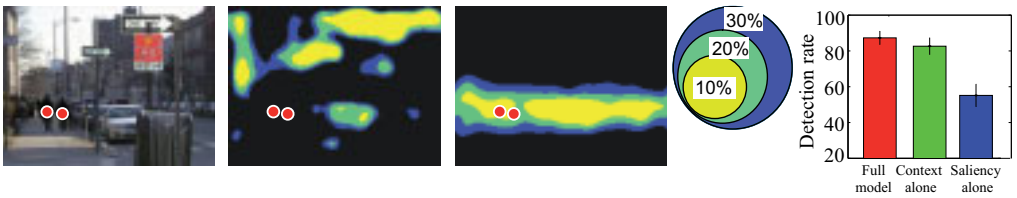


Figure 2.35: Comparison of performance on a detection task between bottom-up saliency, top-down context alone, and the full contextual guidance model by Torralba *et al.* [146]. Performance of the contextual knowledge, while performing well, does not show a statistically significant difference towards the use of top-down context alone. Source: Torralba *et al.* [146].

2.4.5 Utility-Based Vision Systems

In section 2.4.4 on combined bottom-up and top-down saliency vision systems, the necessity to concurrently consider two or more cues to select a region to be observed becomes apparent. While the presented combined bottom-up and top-down saliency vision systems perform cue combination without special consideration of optimality, this problem can also be solved using a formalised utility-theoretical approach.

One example for a utility-based system for vision-guided humanoid walking is proposed by Seara and Schmidt [88, 89]. The approach is based on the maximisation of the predicted visual information gained by observing a certain region in the environment. Visual information is determined by two competing objectives, obstacle avoidance and self-localisation.

The observation of relevant regions is highly task-dependent and requires an adaptation towards the current environment to ensure an optimal application of the available sensor resources. The system proposed by Seara and Schmidt [88, 89] consists of three major modules: information management, task-specific gaze evaluation, and decision making strategy (cf. Fig. 2.36).

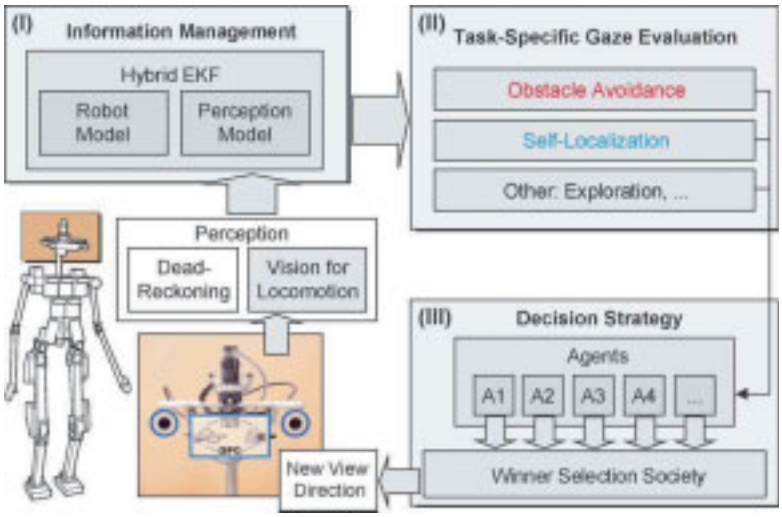


Figure 2.36: Schematic overview of the task-dependent gazing strategy proposed by Seara and Schmidt [88, 89]. Source: Seara and Schmidt [89].

Information Management The tasks of the presented information management module are precise pose estimation and coherent environment map update. For this, a coupled hybrid extended Kalman filter (EKF, cf. Fig. 2.36) presented by Seara *et al.* [149] is employed. The latter uses a models of the biped robot as a physical object, the walking process, and the visual perception with a stereo-camera pair.

Task-Specific Gaze Evaluation Information is quantified using Shannon's Information theory, where

> *"... information is a measure of the decrement of uncertainty".* (Shannon [150])

An efficient resource allocation requires to act in a task-oriented manner and to adapt its attentional strategy to the current situation. At the same time, it is not necessary to minimise all uncertainties simultaneously. Both obstacle avoidance and self-localisation feature a model of incertitude and a predictive gaze-control strategy. This allows the mapping of view directions to incertitudes, that can be used in the decision making process.

Decision Making Strategy The decision making concept proposed by Seara and Schmidt [88, 89] is based upon utility theory, which is closely related to game theory established by von Neumann and Morgenstern [74]. A winner selection society is established, which consists of a group of different agents (cf. Fig. 2.37). These agents propose their respective favourite view directions and their quantitative desire for that direction.

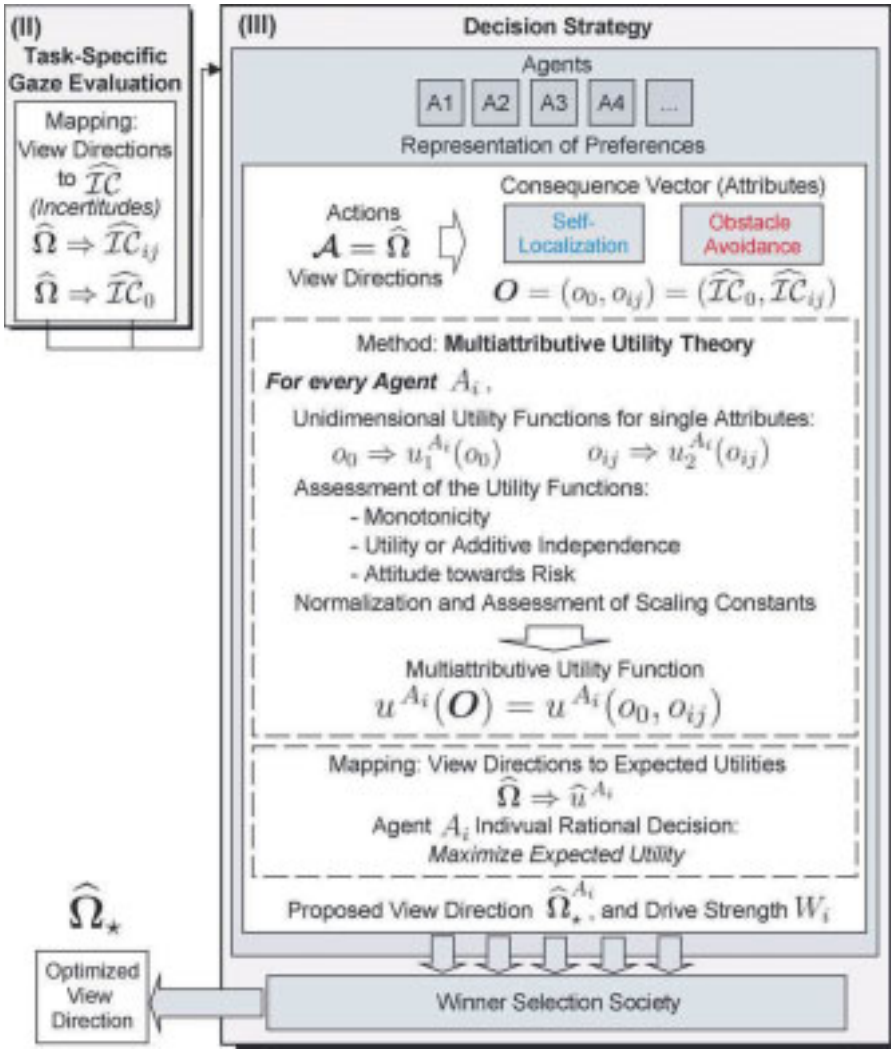


Figure 2.37: Winner selection society proposed by Seara and Schmidt [88, 89]. The figure shows modules (II) and (III) from the schematic overview in Fig. 2.36. Source: Seara and Schmidt [89].

The agents used in the winner selection society feature different strategies: the first group of agents represents its preference ranking using a weighted addition of obstacle avoidance and self-localisation. A second group uses a multiplication of preferences for the ranking. The third group of agents establishes a conservative ranking by the use of a risk averse utility function. Finally, a fourth group of linear approximation agents presents an extension to risk averse agents by using a risk averse linear combination of the two tasks, obstacle avoidance and self-localisation.

In the winner selection society the optimum decision is determined by combining the agents' proposals and respective desires using a meta-decision maker. This meta-agent

can choose from a range of optimisation strategies:

1. Minimise overall utility loss (cf. Utilitarian utility)
2. Minimise worst utility loss (cf. Egalitarian utility)
3. Maximise overall utility gain (cf. Utilitarian utility)
4. Maximise best utility gain (cf. Elitist utility)

The different optimisation strategies are evaluated by Seara and Schmidt [88, 89]. There, a minimisation of overall utility loss shows the best results, followed by the other optimisation strategies in the order as indicated in the above enumeration.

2.4.6 Discussion of Active Vision Systems

The literature on active vision systems shows that a variety of approaches is possible. Below, the transfer of biological concepts to computer vision algorithms and the scopes and properties of the presented methods are discussed.

Transfer of Biological Concepts

Active vision is a field of intensive investigation and has brought forward a multitude of methods, a large fraction of which is inspired by the HVS. A transfer of these approaches is possible as automotive vision systems operate in a road-traffic environment designed to visually provide a human driver with all necessary information. However, a number of differences between the HVS and a technical active vision system be must considered for an implementation into a computer vision system.

During its pre-attentional phase, the HVS performs gaze shifts towards objects that 'pop-out' (cf. Treisman [98]) of their environment. It is accordingly argued by Hubel [117] that the HVS finds it difficult to shift the gaze towards a region lacking local contrast, which can also be seen in Fig. 2.38.

For the human eye, the avoidance of areas with little texture is advantageous, as the area of highest visual acuity (fovea centralis) is as little as 300 μm, which corresponds to an aperture of 1° (cf. Schmidt and Lang [108]). Considering that the usual distance between two communicating persons is 1.2 m (cf. Argyle [151]), the field of acute vision has a diameter of only $1.2\ \text{m} \cdot \tan(1°) \approx 21$ mm.

This small field of vision necessitates saccadic eye movements over the counterpart's face to perceive eyes, nose, mouth, and other significant facial features (cf. Fig. 2.38) which

Figure 2.38: Saccade tracks of the human visual system when observing the respective image. Dots indicate a fixation in between two saccades. Source: Hubel [117].

are then combined into a holistic representation of the face inside the visual cortex. For the human eye to instantly perceive an entire face in high visual acuity, a distance of more than 10 m is necessary.

As opposed to the human eye, cameras usually have a much larger aperture angle. The fixed camera used in our test vehicle has a horizontal aperture angle of 40°, the pan-tilt-zoom (PTZ) camera a variable horizontal aperture angle of 2.8° to 48°, allowing observation of entire cars at a distance of 2 m from the camera. Moreover, object detection and classification methods (cf. section 2.2) are usually designed to analyse the entire object at once.

The difference between biological saccadic perception and camera based holistic perception has to be considered when transferring a biological concept towards a computer vision concept. For example, saliency concepts based upon 'pop-out' characteristics are prone to shifting the centre of a region of interest onto the object's outline, whereas a region of interest centred on the middle of the object is preferable for holistic classification methods.

Scopes and Properties

It must be noted that all discussed active vision concepts, with the exception of Seara and Schmidt [88, 89], lack a formalised Pareto efficient method for multiobjective region selection. However, any solution lacking Pareto efficiency is necessarily suboptimal. The cue-combination strategies of the integrated model by Navalpakkam and Itti [137], the goal-directed search by Frintrop [103], and the contextual guidance model by Torralba *et al.* [146] appear to be Pareto efficient. However, this property is neither explicitly intended, nor claimed in the respective publications. Koene *et al.* [106] use a weighted summation of gaze shift vectors, which can lead to suboptimal decisions for multiple opposing gaze

shift vectors. Contrary to these, Seara and Schmidt [88, 89] describe a formalised Pareto efficient decision method, yet the proposed system does not use unsupervised information such as saliency which presents an essential input for highly reactive systems. An overview of the scopes and properties of all discussed active vision methods is given in Tab. 2.5.

Method	Saliency	Object recognition	Pareto efficient
Centre-surround [99, 100, 102]	yes	no	no
Affine invariant [104]	yes	no	–
Superior colliculus [106]	yes	no	no
Statistical rarity [118, 119]	yes	no	–
Surprise [120, 121]	yes	yes	–
Optimal cue selection [136]	no	yes	–
Excitation / inhibition [103, 138]	no	yes	–
Trained cascades [52, 139]	no	yes	–
Focused vision [105]	no	yes	–
Integrated model [137]	yes	yes	presumably
Goal directed search [103]	yes	yes	presumably
Contextual guidance [146]	yes	yes	presumably
Utility based [88, 89]	no	yes	yes

Table 2.5: Scopes and properties of discussed active vision methods. For every method it is stated whether an unsupervised saliency measure is calculated, an object recognition is performed, or both. For methods performing candidate region selection the Pareto efficiency of the method is stated.

From the scopes and properties of existing methods listed in Tab. 2.5 we infer the need for an efficient resource allocation system integrating both unsupervised saliency and object recognition in a formalised Pareto efficient candidate region selection process.

Chapter 3

Sensor Level

This chapter provides a description of the sensors used in our proposed system, which are only a selection of the sensors available on the test-vehicle presented in section 1.2.3. The differential global positioning system used is described in section 3.1, followed by video cameras in section 3.2. Two range sensors, a single-beam laser scanner (section 3.3), and a photonic mixer device (section 3.4), are described. A discussion of sensor level modules and an overview about the individual sensor specifications is given in section 3.5.

3.1 Differential Global Positioning System

The ego-vehicle's global position, dynamics, and local time are determined by the differential global positioning system (DGPS) with high accuracy. The system's accuracy varies with the availability of an DGPS broadcast, increasing the standard GPS accuracy of 10 m to 15 m to a DGPS accuracy of circa 0.2 m to 0.3 m according to Xu [152].

Information provided directly by the DGPS receiver are the ego-vehicle's position (geographical latitude p_{lat} and longitude p_{lon}), velocity v, direction φ, and the current coordinated universal time (UTC) t which are combined in a measurement vector $\vec{\tau}_{meas}$.

$$\vec{\tau}_{meas} = \begin{pmatrix} p_{lat} \\ p_{lon} \\ v \\ \varphi \\ t_{UTC} \end{pmatrix} \tag{3.1}$$

For the example scene shown in Fig. 3.1 the measurement vector $\vec{\tau}_{meas}$ is

$$\vec{\tau}_{meas} = \begin{pmatrix} 48°46'03''N \\ 11°25'57''E \\ 0.0\frac{m}{s} \\ 247° \\ 11:43:32 \end{pmatrix}$$

3.2 Video Cameras

Our proposed system is equipped with two video cameras, a fixed camera and a pan-tilt-zoom (PTZ) camera. A third, fixed colour video camera (see Fig. 3.3a) is used as a reference sensor and does not feed into the sensor data fusion framework ADTF.

The PTZ camera is a colour camera (AXIS 231D+ [153], mounted on the test vehicle's roof) featuring a zoom-independent resolution. Its movement ranges are 360° for panning and 0°-90° for tilting at a rotational velocity of 360°/s in both directions. Colour is detected by the camera using a standard Bayer pattern (cf. Lian *et al.* [154]). Our traffic participant detectors and classifiers use grayscale information, therefore the colour-image is converted into a luminance map.

The fixed video camera is a grayscale camera (MatrixVision mvBlueFOX-120, mounted behind the windscreen, see Fig. 3.3a). An example image acquired using the fixed camera can be seen in Fig. 3.1.

Figure 3.1: Fixed grayscale camera video image of an example scene.

3.3 Laser Scanner

We use a single-beam time-of-flight laser scanner (Sick LMS291, cf. Ye and Borenstein [155]) mounted on the test vehicle's radiator grille.

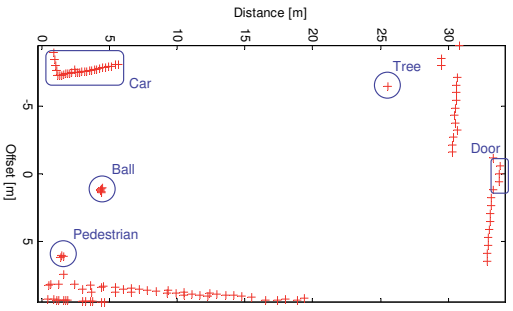
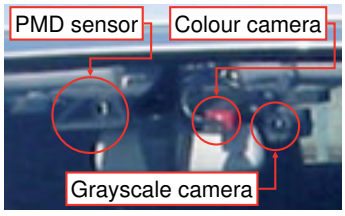
a)  b) 

Figure 3.2: a) Laser scanner mounted to the test vehicle's radiator grille. b) Laser scanner readings of the scene shown in Fig. 3.1.

The laser scanner readings shown in Fig. 3.2 contain objects such as a pedestrian and a car situated outside the video camera's aperture angle.

3.4 Photonic Mixer Device

The photonic mixer device (PMD) sensor is a 3-D camera mounted behind the windscreen next to the fixed video camera. The 3-D camera operates using the phase shift of returning light towards a set of active light sources (e.g. Fardi *et al.* [58]), which emit a modulated wave front of high-energy infrared light.

a) 

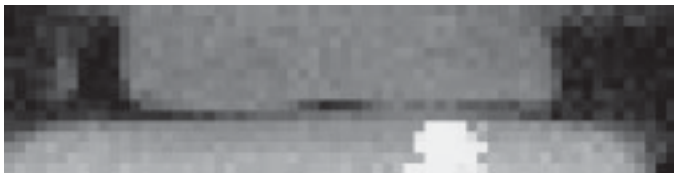
b) 

Figure 3.3: a) Cameras mounted behind the windscreen: (from left to right) PMD sensor, reference (colour) camera, and grayscale camera. b) PMD range image of the scene shown in Fig. 3.1.

It can be seen in Fig. 3.3 that the range information provided by the PMD sensor contains a considerable amount of noise as compared to the laser scanner measurements shown in Fig. 3.2. However, a three-dimensional range map is obtained as opposed to a single-beam range profile.

3.5 Discussion of Sensors

In this chapter the sensors used in our proposed system are described. Tab. 3.1 provides an overview of the used sensors' properties.

Property	Fixed camera	PTZ camera zoomed	PTZ camera wide	PMD sensor	Laser scanner
Modality	luminance	luminance	luminance	range	range
Maximum range	–	–	–	20 m	30 m
Aperture angle	40° × 30°	2.8° × 2.1°	48° × 36°	55° × 14°	180° × 1°
Resolution	640 × 480 px	704 × 576 px	704 × 576 px	64 × 16 px	181 samples
Acuity	16.0 px/°	181 px/°	10.6 px/°	0.86 px/°	1 sample/°
Sample rate	25 Hz	25 Hz	25 Hz	≥50 Hz	75 Hz

Table 3.1: Overview of used exteroceptive sensors' properties. The synchronisation frequency of the ADTF is 25Hz.

From the properties in Tab. 3.1 it can be seen that our sensor system relies on two modalities: luminance and range. The maximum ranges of our selected sensors are 30 m and less. Both properties have implications on the active vision system and are discussed in sections 3.5.1 and 3.5.2 below. Apart from these properties, the variety of aperture angles and acuities is considerable ranging from 1 sample/° and an 180° × 1° aperture for the laser scanner to 181 px/° and a 2.8° × 2.1° aperture for the zoomed PTZ camera.

3.5.1 Sensor Modalities

In the data level of our proposed system both luminance and range are used as exteroceptive modalities besides the use the ego-vehicle's global position and velocity. Luminance information is acquired using two cameras and range information is acquired by both a PMD sensor and a laser scanner. A DPGS module acquires the ego-vehicle's global position and dynamics.

In our proposed system data from short-range radar, long-range radar, and ultrasonic sensors is discarded. Ultrasonic range information is discarded due to its limited detection range of less than 2 m, which is helpful for parking scenarios, but less so for safety-related driver assistance systems. To discard radar information is an ambivalent decision. It can be argued that range information and dynamics information about other traffic participants acquired by Doppler-radars is indispensable, especially for determining time-to-collision, which is done in our proposed system. However, including radar information also has a number of disadvantages such as increasing system complexity, limited angular resolution, and the necessity to perform multi-sensor track-to-track fusion.

First, using additional sensors increases complexity as characteristic problems caused by each sensor type have to be accounted for in the proposed system. Second, radar sensors have a limited angular aperture angle and resolution. For the determination of candidate regions this provides only limited value. Third, each radar provides a pre-tracked list of targets, which necessitates a multi-sensor track-to-track fusion algorithm. Our investigations in Matzka and Altendorfer [15, 16] show that sensor information is correlated due to the use of a common dynamics model, and therefore must not be fused with the Kalman-Tracker presently used in the ADTF sensor data framework.

3.5.2 Sensor Ranges

The maximum ranges of our sensors are less than 30 m, mainly due to the discarding of radar information which extends the sensors system's horizon beyond 100 m. This limits the time available to allocate resources and thus the relative velocities at which other vehicles can reliably be detected. Assuming that both the ego vehicle and a vehicle on the opposing lane of a country road move at 30 $\frac{m}{s}$, the relative velocity is $v_{rel} = 60\ \frac{m}{s}$. In the case of an accident, the time-to-collision at maximum sensor range is

$$\frac{d_{max}}{v_{rel}} = \frac{30\ m}{60\ \frac{m}{s}} = 0.5\ s$$

which then falls into the pre-crash period rather than the resource allocation period (cf. Tab 7.12).

In summary, the sensors used in our proposed system are similar to the sensors used for autonomous driving systems' sensors as discussed in section 2.1.1, with the exception of the multi-beam laser scanners used on autonomous vehicles, which is emulated by the use of a PMD sensor to some degree. The limited detection range of our proposed sensor system therefore leads to a focus on traffic participants with low relative velocities, which essentially is traffic driving in the same direction.

Chapter 4

Data Level

Data level representations are acquired using low-level data processing methods on sensor data. This level of abstraction is highly reactive as it is feasible to calculate all data-level features in real-time using computationally inexpensive algorithms. In this chapter, each data-level module is discussed with respect to the available sensor data and the required processing steps. The used sensors are installed at different positions on the vehicle and use different internal coordinate systems. Therefore the used coordinate systems and coordinate transformations are discussed in section 4.1. In section 4.2 the position and velocity of the ego-vehicle are presented. Luminance information is described in section 4.3, range information in section 4.4, and finally motion estimation in section 4.5. A discussion of data level modules in section 4.6 concludes this chapter.

4.1 Coordinate Systems

In order to represent measurements from different sensors, two transformable coordinate systems, plan view and perspective view, are used.

4.1.1 Plan View

Laser scanner measurements are provided by the sensor using an ego-vehicle centred coordinate system with the origin at the vehicle's centre at road level (cf. Fig. 4.1). This type of data is represented using a plan view, an orthographic projection of the three dimensional environment. Coordinate axes used to describe positions in this system are x (longitudinal), y (lateral), and z (elevation above ground plane) and are usually expressed

with a position vector $\vec{p}$.

$$\vec{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \tag{4.1}$$

Fig. 4.1 shows a car model with a plan view coordinate system. In accordance with the ISO 8855:1991 standard [156], the coordinate system's origin is at vehicle's centre projected down to ground plane level, with the x-axis parallel to the vehicle's longitudinal axis.

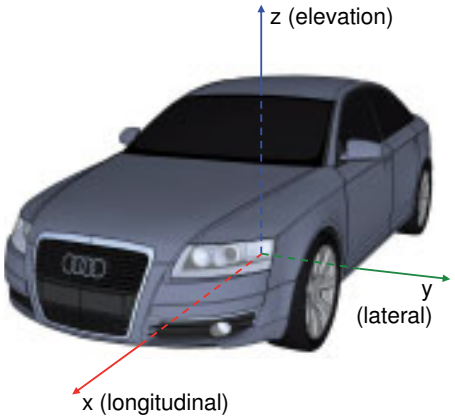


Figure 4.1: Ego-vehicle centred coordinate system for plan-view representations such as radar targets or laser scanner measurements. The coordinate system's origin is the vehicle's centre projected down to ground plane level, with the x-axis parallel to the vehicle's longitudinal axis according to ISO 8855:1991 standard [156]. Source: Audi AG.

4.1.2 Perspective View

Sensors such as video cameras or a PMD sensor acquire a perspective view of the environment. Each measured pixel $\vec{p}(i, j)$ can be assigned corresponding zenith and azimuth angles as well as a radial distance r, if range information is available.

$$\vec{p} = \begin{pmatrix} i \\ j \\ r \end{pmatrix} \tag{4.2}$$

Fig. 4.2 shows a car model with a perspective view coordinate system centred at the respective sensor.

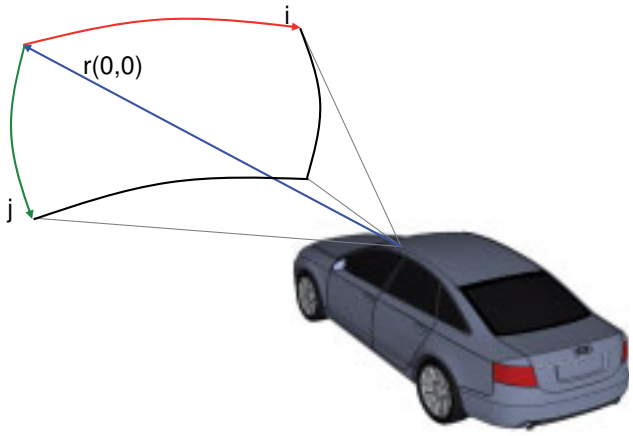


Figure 4.2: Coordinate system for perspective representations such as video or PMD sensor measurements. The centre of the coordinate system is the respective sensor. Source: Audi AG.

4.1.3 Coordinate Transformation

In order to transform coordinates from one representation into the other, the positions, orientations and aperture angles of the respective sensors have to be known. In Audi's sensor data framework ADTF, this information is provided for every sensor, enabling coordinate system transformation within the sensor framework. These files also contain registration and calibration information to be used for sensor data fusion. For our system, coordinate transformations are performed within the ADTF using standard coordinate transformation methods.

The main coordinate transformation in our proposed system is the transformation from plan view into perspective view, which is commonly referred to as a camera transformation (cf. Riley *et al.*[157]), and is given in Eq. 4.3.

$$\vec{p'} = \begin{pmatrix} i' \\ j' \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta_x) & \sin(-\theta_x) \\ 0 & -\sin(-\theta_x) & \cos(-\theta_x) \end{pmatrix} \cdot \begin{pmatrix} \cos(-\theta_y) & 0 & -\sin(-\theta_y) \\ 0 & 1 & 0 \\ \sin(-\theta_y) & 0 & \cos(-\theta_y) \end{pmatrix} \cdot \begin{pmatrix} \cos(-\theta_z) & \sin(-\theta_z) & 0 \\ -\sin(-\theta_z) & \cos(-\theta_z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} \right] \tag{4.3}$$

where $\vec{p'}$ is the 3-D position relative to the camera's coordinate system dependent upon

the camera's position $c_{x,y,z}$ and orientation $\theta_{x,y,z}$ of a coordinate $p_{x,y,z}$ in the plan view coordinate system. In our test vehicle the fixed camera's orientation $\theta_{x,y,z}$ is approximately parallel to the x-axis of the plan view coordinate system

$$\theta_x = \theta_y = \theta_z \approx 0$$

therefore our camera transform for the fixed camera simplifies to

$$\begin{pmatrix} p_{i'} \\ p_{j'} \\ p_r \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} \tag{4.4}$$

As both i', and j' are still given in metres, a transformation towards pixel values i and j must be performed

$$i = \left(\frac{\delta \cdot i'}{r}\right),\ j = \left(\frac{\delta \cdot j'}{r}\right) \tag{4.5}$$

where the focal length δ is the distance to a virtual projection plane (cf. Carlbom and Paciorek [158]). In order to transform i' and j' into pixel values i and j, δ is determined to be

$$\delta = \frac{i_{max}}{2 \cdot \tan(\alpha_{hor})} \tag{4.6}$$

where α_{hor} is the horizontal aperture angle of the camera.

4.2 Position and Velocity of Ego-Vehicle

The measurement vector $\vec{\tau}_{meas}$ provided by the DGPS is used to update the ego-vehicle's state vector $\vec{\tau}$

$$\vec{\tau} = \begin{pmatrix} p_{lat} \\ p_{lon} \\ v \\ \varphi \\ t_{local} \end{pmatrix} \tag{4.7}$$

where p_{lat} and p_{lon} are the ego-vehicle's global latitudinal and longitudinal position, v the ego-vehicle's absolute velocity, φ the ego-vehicle's direction, and t the local time. Local time is determined by adding or subtracting the local time-shift towards the coordinated universal time (UTC) acquired by the DGPS system using p_{lat} and p_{lon} to determine the current time-zone.

4.3 Luminance

Luminance information is acquired by video cameras and by the PMD sensor. The fixed camera in our test vehicle acquires video frames at a resolution of 640 × 480 px. These frames are resampled to be used as low-resolution and high-resolution intensity representations for our evaluation. The downscaled image dimensions are 320×240 pixels and span the same region as the original image. Upscaling is not performed on the whole image but only for a single region of 160×120 pixels, which is then upscaled to a 320×240 image using a high-quality *Lanczos3* resampling method described by Pharr and Humphreys [159].

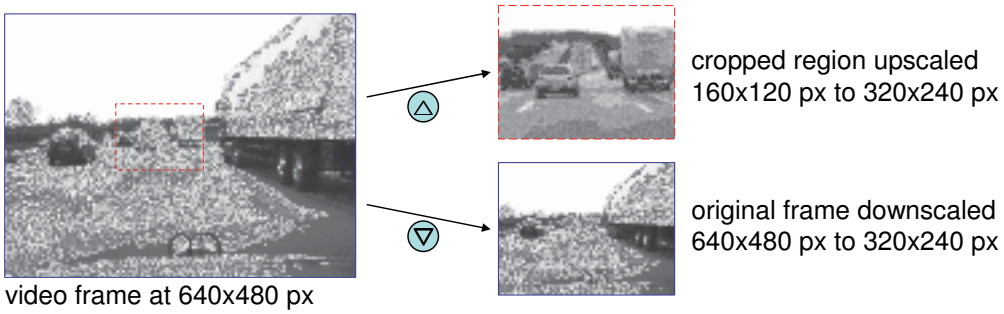


Figure 4.3: High-quality upscaling (cropped region from 160×120 px to 320x240 px) and downscaling (original frame from 640×480 px to 320×240 px) of a 640×480 px video frame.

The output matrix of the luminance values $l(i, j)$ in an image is represented as a luminance matrix L.

$$L = \begin{pmatrix} l(0,0) & l(0,1) & \cdots & l(0,j_{max}) \\ l(1,0) & l(1,1) & \cdots & l(1,j_{max}) \\ \vdots & \vdots & \ddots & \vdots \\ l(i_{max},0) & l(i_{max},1) & \cdots & l(i_{max},j_{max}) \end{pmatrix} \tag{4.8}$$

4.4 Range

Range is determined using both a PMD sensor and a single-beam laser scanner. The sensors use different coordinate systems, therefore a coordinate transformation towards a common coordinate system is required for range data fusion. In our test vehicle, the registration information of all sensors is available and the necessary transformation is performed by the ADTF.

The output matrix of the range map acquired by the PMD sensor is represented as a range matrix R_{PMD} using a polar coordinate representation.

$$R_{PMD} = \begin{pmatrix} r(0,0) & r(0,1) & \cdots & r(0,j_{max}) \\ r(1,0) & r(1,1) & \cdots & r(1,j_{max}) \\ \vdots & \vdots & \ddots & \vdots \\ r(i_{max},0) & r(i_{max},1) & \cdots & r(i_{max},j_{max}) \end{pmatrix} \tag{4.9}$$

The range profile acquired by the laser scanner is represented in a polar coordinate range vector R_{LS}. The laser scanner provides one range measurement $r(n)$ for every 1° in the range n=[0°,180°], resulting in a total of 181 range readings.

$$R_{LS} = \big(r(0),\ r(1),\ \cdots,\ r(180)\big) \tag{4.10}$$

4.5 Motion

From the two range representations R_{PMD} and R_{LS}, we determine two motion representations relative to our ego-vehicle in our proposed system: range profile motion and motion vector maps.

4.5.1 Range Profile Differentiation

A simple range profile motion representation is calculated by numerically differentiating the laser scanner's range measurement. For every range value $r_k(n)$ at cycle k, the relative range profile motion $\tilde{v_k}(n)$ is determined to be the median of m range motion values using Eq. 4.11.

$$\tilde{v_k}(n) = \text{median}\left(\frac{r_k(n) - r_{k-1}(n)}{t_k - t_{k-1}};\ \ldots\ ; \frac{r_{k-m+1}(n) - r_{k-m}(n)}{t_{k-m+1} - t_{k-m}}\right) \tag{4.11}$$

where n is the degree value, and t_k the time-stamp of the respective cycle number. The use of multiple velocity values is also uncritical from a real-time perspective due to the high sample rate of 75 Hz of the laser scanner. An example for possible laser readings over time is given in Tab. 4.1.

k	t_k [ms]	$r_k(0)$ [m]	$r_k(1)$ [m]	$r_k(2)$ [m]
0	0	5.03	5.00	5.02
1	65	5.01	5.02	4.91
2	132	5.02	9.84	4.83
3	198	5.04	9.96	4.74

Table 4.1: Example range readings for n=0°..2° and k=0..3 acquired by a laser scanner. The values for 0° demonstrate normal measurement noise, the values at 1° show a step caused by an object exiting the sector between k=1 and k=2. The values for 2° show an object coming closer to the laser scanner for every cycle.

In Tab. 4.2 the individual velocity values $v_k(n)$ and the median velocity $\tilde{v}$ are calculated using Eq. 4.11.

k	$t_k - t_{k-1}$ [ms]	$v_k(0)$ $[\frac{m}{s}]$	$v_k(1)$ $[\frac{m}{s}]$	$v_k(2)$ $[\frac{m}{s}]$
1	65	-0.31	+0.31	-1.69
2	67	+0.15	+71.9	-1.19
3	66	+0.30	+0.30	-1.36
$\tilde{v}_3(n)$		+0.15	+0.31	-1.36

Table 4.2: Individual relative velocity calculations $v_k(n)$ gained by differentiation and median velocity $\tilde{v}_3(n)$ for the example measurements given in Tab. 4.1.

From Tab. 4.2 it can be seen that the median filtering of the velocity is able to remove the velocity outlier $v_2(1)$. This is an important property, as outliers caused by temporal step edges in the range profile present a problem as no object tracking is applied.

As an example for range profile differentiation, a motorway sequence with overlayed time-to-collision information estimated using laser range data on video data is shown in section 5.5. There, an example frame of the motorway sequence can be seen in Fig. 5.33 on page 138.

Both Tab. 4.2 and graphical TTC representation in Fig. 5.33 show that the median velocity $\tilde{v}$ is a robust indicator of the relative motion of a surface towards the laser scanner. Due to the laser scanner's mounting position on the vehicle's radiator grille, this relative motion can be used to estimate time-to-collision with an object, which is discussed in section 5.5.

4.5.2 2-D and 3-D Motion Vector Maps

Motion vector maps can be determined using both intensity and range image sequences. In the following, a method to perform 3-D translational motion estimation adapted from a fast 2-D motion estimation technique (PMVFAST, proposed by Tourapis *et al.* [160]) presented in Matzka *et al.* [13] is discussed.

Related Work on Motion Estimation

Estimating 3-D motion or optical flow fields from range images is a known problem in the literature. For example, an evaluation of 3-D motion estimation algorithms is given in Eggert [161]. Many 3-D motion estimation approaches are based upon finding correspondences. These correspondences can be considered both local as in Chaudhury *et al.* [162] or global by solving a total least squares framework as proposed by Spies *et al.* [163]. The resulting flow field of the latter method is dense, yet the complexity is high and real-time computation is not feasible with current automotive ECUs.

A correspondenceless approach was pursued by Liu and Rodrigues [164], based upon the cross matrix to estimate the motion parameters. Jiang *et al.* [165] use the shift of previously segmented surfaces in a range image for motion estimation. This approach is restricted to small relative motion between the camera and the scene and the segmentation process itself is complex.

Apart from the cited work on 3-D motion estimation, 2-D optical flow is a major topic of interest. Most of the 2-D motion estimation algorithms used in video-encoders are designed to be computationally efficient, which is also a constraint for real-time motion estimation. However, to estimate 3-D motion in range images under real-time constraints, neither 2-D motion estimation based on difference measures, nor 3-D motion estimation algorithms with high complexity can be used. Therefore, we propose the extension of a 2-D motion estimation algorithm for use on range images.

2-D Motion Estimation using PMVFAST

The Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) proposed by Tourapis *et al.* [160] is a block based motion estimation technique based upon MVFAST [166], which is an essential part of several video-coding standards, such as MPEG-1/2/4. In Tourapis *et al.* [160], PMVFAST is shown to be faster than other motion es-

timators while retaining a motion estimation quality comparable to a significantly slower full search algorithm.

PMVFAST uses a diamond search (DS) pattern as shown in Fig. 4.4a). Beginning in the centre, the (0,0) motion vector (MV) is the initial starting point. The search path then meanders circularly around the centre, performing a full orbit each time before increasing its search distance up until the maximum search distance.

At each point on the search path, a block in the previous frame is matched against a block in the current frame. The block in the current frame is shifted by the (i,j) values of the search path. The quality of the match is determined by a distortion measure. A widely used distortion measure is the sum of absolute differences (SAD, Eq. 4.12), which omits the multiplications necessary for mean squared error but has a similar performance according to Tourapis *et al.* [160]. We use a block size of 5×5 px, resulting in 25 summations per comparison. Motion vectors are not calculated for every pixel, instead a regular grid is used.

$$SAD_{DS}(v_i, v_j) = \sum_{m,n \in DS} |I_k(i+m, j+n) - I_{k-1}(i+v_i+m, j+v_j+n)| \tag{4.12}$$

The search for the minimum SAD is performed with two differently sized diamonds in Tourapis *et al.* [160]. The expected magnitude of motion is estimated by examining three neighbouring MVs at $(i-1, j)$, $(i, j-1)$, $(i+1, j-1)$, the previous MV at (i_{k-1}, j_{k-1}), and the median MV (cf. Fig. 4.4b). The mean value for these MVs is then used as an estimate for the current MV.

If the estimated MV for (i,j) is small (i.e. $|MV| \leq 1$ px), a small 2×2 px search

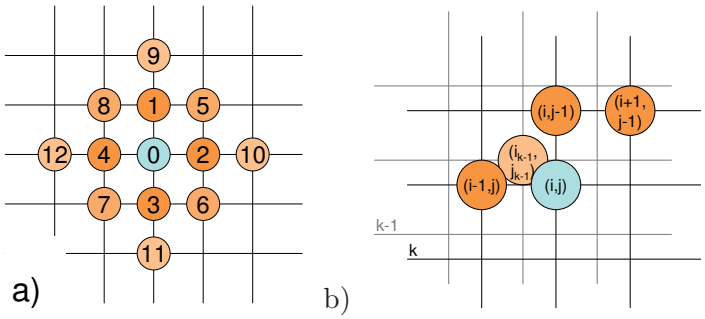


Figure 4.4: Fig. a) shows a diamond search pattern used for 2-D motion estimation. b) Neighbouring motion vectors, both spatial and temporal, are used to predict the current motion vector.

diamond is used with the (0,0) MV as its centre. If the MV is estimated to be (i.e. $|MV| \leq 3$ px), a larger 3×3 px diamond is used, again with (0,0) MV as starting point. In the case of high estimated motion (i.e. $|MV| > 3$ px), the small 3×3 px diamond is used with the estimated MV as its centre.

If the examined distortion is below a predetermined threshold, no further matching is done. Otherwise, the DS is performed and the displacement featuring the minimum distortion is chosen as the centre point for the next cycle. The search algorithm terminates if the centre of the search diamond is also the displacement with minimum distortion. This concept is designed for use on intensity images, yet in range images distance information is represented by intensity. On convex surfaces, such as a sphere, this induces a difference-based 2-D motion estimation to detect a concentric outward motion if the distance is decreasing (assuming that small distances are represented by a high intensity), and a converging motion if the distance is increasing (cf. Fig. 4.5).

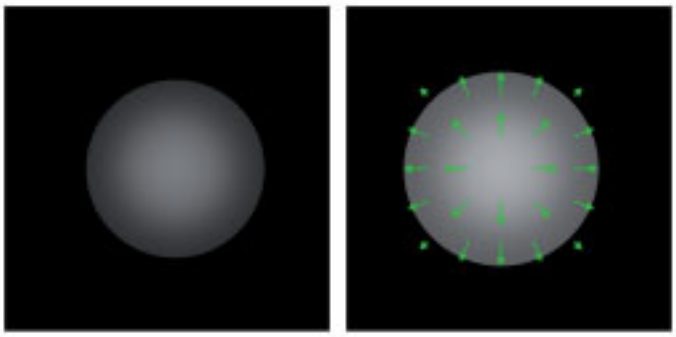


Figure 4.5: Example for concentric, and converging (not shown) motion vector effect that appears if a sphere is changing its relative distance.

The above behaviour does not affect the quality of MPEG motion estimation, since a video codec's objective is to maximally reduce the video's bit rate while having as little visible quality loss as possible as opposed to calculating exact MVs. For range images this effect leads to the necessity to consider depth motion to get accurate motion vectors.

Extending Diamond Search for use on Range Images

We extend the idea of using a diamond shaped search path towards a 3-D translational motion estimation from range images. The least complex diamond shape in 3-D is a regular octahedron which is referred to as *point cut search* (PCS) path.

The PCS path is expanded incrementally, adding new layers around the origin in a

point cut shape. The first layer has a distance of 1.0 to the origin and consists of the six permutations

$$(1,0,0),\ (0,1,0),\ (0,0,1),\ (-1,0,0),\ (0,-1,0),\ (0,0,-1)$$

with varying signs.

The following base coordinates are (1,1,0), (1,1,1), (2,0,0), (2,1,0), (3,0,0) etc. All base coordinates are then permutated (for a maximum of six permutations if all values are unique) with changing signs for every value (for a maximum of eight sign combinations if no value is zero). An illustration of the PCS path building process is given in Fig. 4.6a-c.

Both PMVFAST and PCS realise horizontal and vertical displacements by shifting the observation window in the actual frame horizontally and vertically. In PCS, displacements in distance in range images are represented as changes of intensity. Therefore, by adding or subtracting the value corresponding to the range displacement to the intensity values in the observation window, a displacement in distance can be modelled (see Eq. 4.13).

$$SAD_{PCS}(v_x, v_y, v_z) = \sum_{i,j,k \in PCS} \left| I_k \begin{pmatrix} x+i \\ y+j \end{pmatrix}' - I_{k-1} \begin{pmatrix} x+v_x+i \\ y+v_y+j \end{pmatrix}' + v_z + k \right| \quad (4.13)$$

In Eq. 4.13 the use of SAD as a distance measure is motivated by its low computational cost and its quality as a measure which is comparable to computationally more expensive methods such as using a correlation measure. As for PMVFAST, the search terminates when the centre point of the PCS is also the point with minimum SAD or when the maximum number of iterations is reached. As an example for the performance of both PMVFAST and PCS on range images Fig. 4.7 shows the motion vectors for objects with

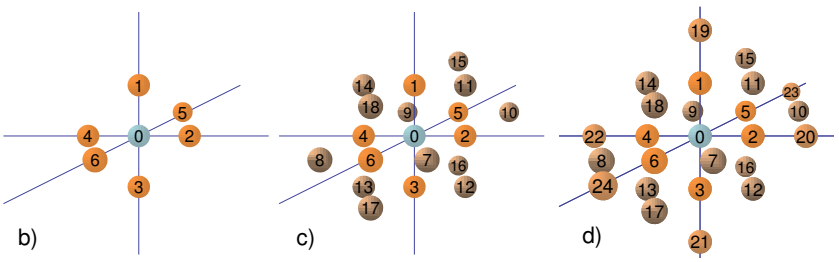


Figure 4.6: An example PCS path building process is shown: a) for all (1,0,0) permutations (*1-6*), b) extends a) with all (1,1,0) permutations (*7-18*), and c) extends b) with all (2,0,0) permutations (*19-24*).

increasing and decreasing distances.

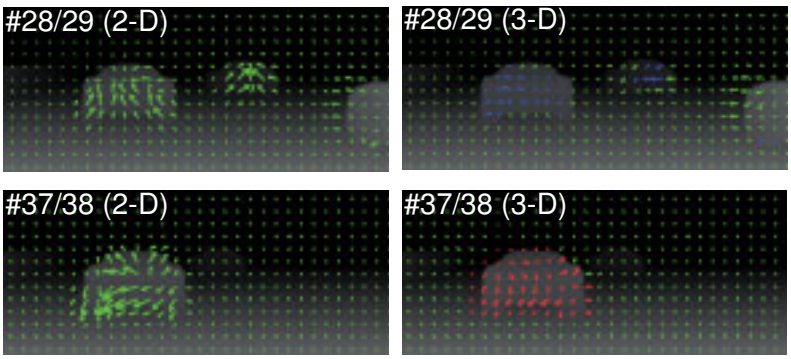


Figure 4.7: Motion vector fields of frame pairs #28/29 (increasing distance to car in front) and frame pair #37/38 (decreasing distance) in the TOR sequence[1]. Motion vector field (2-D) shows the result using a 2-D full search algorithm, whereas (3-D) shows the PCS result. Blue arrows indicate an increasing distance, red arrows a decreasing distance. The background shows the range images on which the motion estimation has been performed.

Evaluation of 3-D Motion Estimaton

Our proposed PCS 3-D motion estimator is implemented using four sequential processing steps shown in Fig. 4.8. First, range images are filtered to remove noise. Second, subsequent filtered range images are searched for correspondences using PCS. Third, resulting motion vectors are filtered to remove outliers. Fourth, the filtered motion vector field is used to predict the motion vectors for the next motion estimation (cf. Fig. 4.4b).

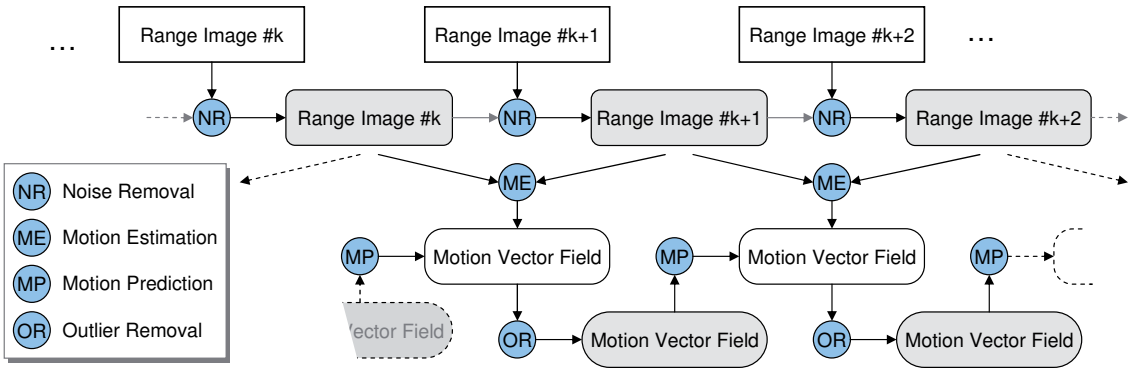


Figure 4.8: Block diagram of the implemented PCS motion estimator. Circles represent processing / filtering operations that are performed by the motion estimator, while boxes represent different abstraction layers from unfiltered range images to filtered MV fields.

Computational Cost The computational cost of the implemented motion estimator is evaluated using a simulated range image sequence[1] (TOR, cf. appendix A) extracted from *Torcs*[2], an open source racing game. The sequence consists of 155 frames recorded at 15 frames per second and a resolution of 500×220 px. Range is encoded with 8 bit, providing a coarse yet sufficient range resolution.

For each configuration, the average number of comparisons required for each motion vector and the average SAD for the chosen motion vectors are taken as indicators of the computational cost and motion vector field quality respectively. To get a benchmark for these two values, a full search (FS) is used (cf. Tab. 4.3).

	FS	PCS_2	PCS_3	PCS_4	PCS_5	PCS_6	PCS_7
Comparisons per MV	75.52	19.72	22.49	24.70	25.72	26.48	27.10
Mean SAD per MV	33.16	45.46	40.21	37.04	35.69	34.60	33.86
Efficiency measure ($\prod$)	2504.4	897.5	904.3	915.0	918.0	916.1	917.6

Table 4.3: Comparisons per MV and average SAD for motion estimation in the Torcs sequence. A full search (FS) is used as benchmark for the PCS_n with n maximum iterations. The efficiency measure is the product of comparisons per MV and average SAD, lower values are better.

For an evaluation of the computational efficiency of the PCS search strategy the maximum number of iterations to shift the local minimum to the PCS's centre is used. For evaluation, two PCS paths are chosen, the small PCS with a maximum search distance of 2 px and the large PCS with a maximum search distance of 5 px.

Tab. 4.3 shows the performance of the PCS strategy with respect to the maximum allowed number of iterations. The lowest mean SAD of 33.86 for PCS_7 is comparable to the benchmark value $\overline{SAD}_{FS}$ of 33.16, while requiring only 36% of the comparisons.

In order to assess the efficiency of the PMVFAST search strategy, the product of comparisons required for each MV and the mean SAD is a used as an efficiency measure. This product grows with increasing computational cost and distortion, for low computational cost and low distortion the product is small (cf. Tab. 4.3), the latter being true for PCS.

Quantitative Evaluation of Accuracy A comparison of the estimated motion vector fields of a synthetic motion pattern against a ground truth known from the rendering process of the pattern is described below.

[1] The TOR sequence is available online: http://www.matzka.net/vision/html/torcs.html
[2] Torcs is an open source racing game (http://torcs.sourceforge.net) using OpenGL.

Motion Ground Truth The motion pattern consists of two spheres diametrically orbiting around the range image's centre (i,j,r) = (160,120,127) so that the sphere in front occludes the sphere behind it intermittently. The underlying motion function for this pattern is

$$v_k = \begin{pmatrix} \lfloor 80.0 \cdot \sin(\frac{k}{30}) + 160.5 \rfloor \\ \lfloor 60.0 \cdot \cos(\frac{k}{30}) + 120.5 \rfloor \\ \lfloor 80.0 \cdot \cos(\frac{k}{30}) + 127.5 \rfloor \end{pmatrix} - \begin{pmatrix} i_{k-1} \\ j_{k-1} \\ t_{k-1} \end{pmatrix}, \; v_{max} = \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix} \tag{4.14}$$

The resulting range image sequence[3], a subset of which can be seen in Fig. 4.9.

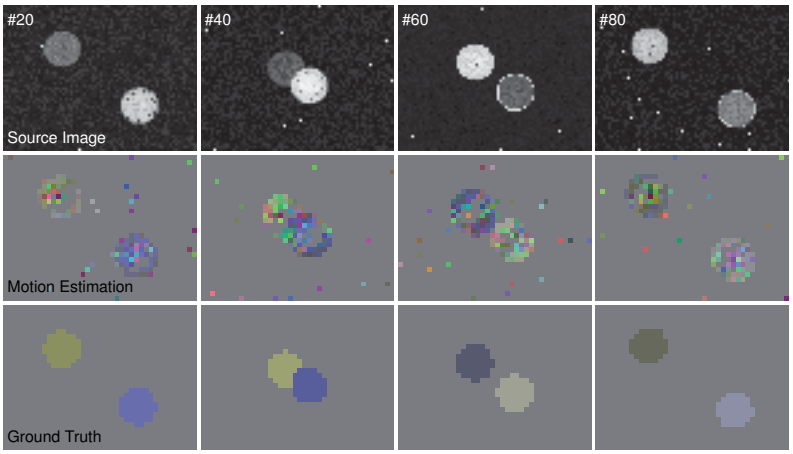


Figure 4.9: Example frames of two spheres diametrically orbiting around the range image's centre. Motion components are RGB-colour coded for v_i(red), v_j(green), and v_r(blue). The source frames are noised using a Gaussian noise with $\sigma = 2.7$.

Noise Removal and Preprocessing Range data sequences acquired by a 3-D camera suffer from a substantial amount of noise. This noise can be reduced by employing a temporal Gaussian filter on the present frame and a number of previous frames. For traffic scenes, temporal filtering over a number of frames may include rotational motion of moving objects, which is not handled well by the algorithm. In this trade-off between noise and rotational motion our algorithm is shown to be more capable of handling noise in range images, therefore only a small number of frames is used for temporal Gaussian filtering.

The measurement noise of our range data sequences acquired with the ego-vehicle's

[3]Available online: http://www.matzka.net/vision/html/orbit.html The sequence contains 200 frames with 320 × 240 px showing the source range image, ground truth, motion estimation and motion vector field (from left to right).

PMD sensor is best characterised as clipped Gaussian noise with $\sigma = 2.7$ range units in the 8 bit range map, as no negative distances or distances above the maximum measurable distance can appear (cf. Fig. 4.10). We superimpose a Gaussian range noise r_{noise} with $0.0 \geq r(i,j) + r_{noise} \geq 255.0$, onto the synthetic range image sequence to simulate the real PMD sensor.

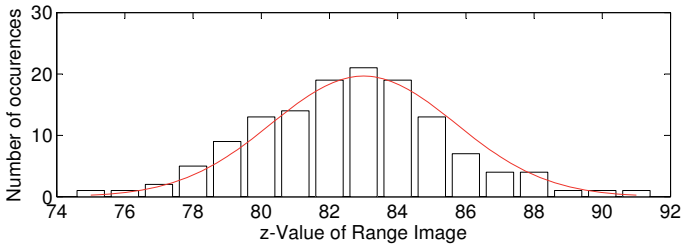


Figure 4.10: Distribution of PMD range measurements of a constant distance over 135 frames (bars). This noise distribution can be approximated by a Gaussian distribution with $\sigma = 2.7$ (red line).

Assuming a Gaussian noise model, using a Gaussian filter considering neighbouring pixels with $0.8 \geq \sigma_{RI} \geq 4.8$ presents suitable preprocessing (cf. Fig. 4.11).

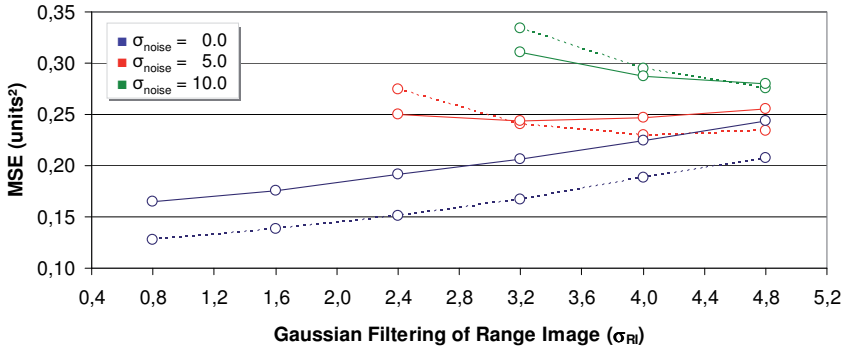


Figure 4.11: Mean squared error of motion vector components for the orbiting movement pattern under influence of Gaussian noise σ_{noise} estimated by PCS_3 (solid line) and FS (dotted line) as compared to ground truth. The range image is processed using a Gaussian filter with σ_{RI}.

In Fig. 4.11, three major effects can be observed. First, if a noise-free range image is processed with a Gaussian filter, the MSE deteriorates be expected. Second, if a noisy range image is processed with a Gaussian filter, the MSE decreases until a point where the range image is quasi noise-free and then shows the same behaviour as a noise-free image (i.e. MSE deterioration for higher standard deviations).

The third observable effect is that PCS has a lower MSE than FS for range images with a high remaining noise after preprocessing. The reason for that is a differing termination

condition. If a high level of noise is present during motion estimation, the correct MV does not necessarily exhibit the lowest SAD value. Using a FS approach, every displacement has the same probability to be selected as the estimated MV, whereas the iterative shifting in PCS increases the probability that a displacement near the initial starting point is selected.

The synthetic scene contains a large fraction of (0,0,0) MVs, therefore an incorrect MV close to an initial (0,0,0) MV starting point does not affect the MSE as much as a large MV, which is more probable to occur using a full search. However, it can be seen in Fig. 4.11 that this effect disappears when a suitable level of filtering is applied, so that the correct MSE exhibits the minimum SAD.

Outlier removal An analysis of the resulting MV fields against ground truth information suggests that the main reason for high MSE values of the estimated motion vector fields is single irregular motion vectors caused by noise in the range image, not generic false motion vector estimation. Suitable methods to achieve noise reduction and outlier removal include Gaussian or median filtering of the MV field.

In Fig. 4.12, MSE values for the same synthetic range image sequence as in Fig. 4.11 when using a Gaussian ($\times$) or median (Δ, 5×5 px) filter are shown.

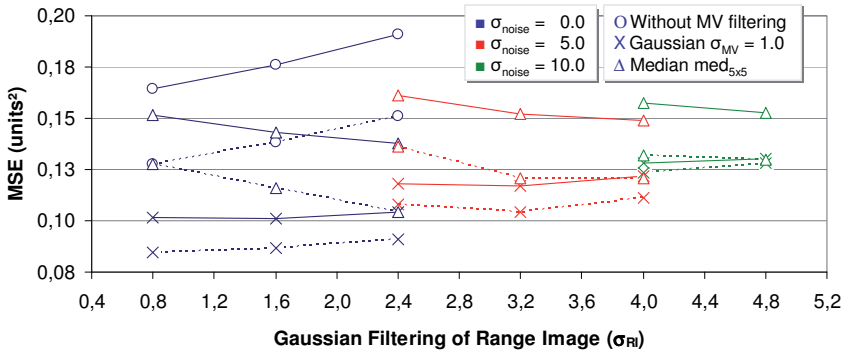


Figure 4.12: Mean squared error of motion vector components for the orbiting movement pattern under influence of Gaussian noise σ_{noise} estimated by PCS_3 (solid line) and FS (dotted line) as compared to ground truth. The source range image is filtered using a Gaussian filter with σ_{RI}. The motion vector is postprocessed using either a Gaussian filter or a median filter.

In can be seen from Fig. 4.12, that the optimum MSE values gained by PCS at different levels of noise in the range images (including no noise) are within a narrow field (that is 0.102 to 0.162). This is an indicator that the algorithm is robust towards noise if both input range images and motion vector fields are suitably filtered. Since the Gaussian

filtering of the motion vector field results in a lower overall MSE than median filtering, the errors in the motion vector fields must be assumed to be the result of a noise process as opposed to systematic outliers. The results are also comparable with the results gained by FS. At the same time, PCS computed the 320 × 240 px range image sequence at 11.8 frames per second (fps) on a standard 2.0 GHz PC, where FS performed at 1.85 fps, thus being more than six times (6.38) slower.

Performance on Data acquired with a 3-D Camera In addition to synthetic range image sequences, the proposed algorithm is evaluated using data acquired by a PMD sensor. The 3-D sensor acquires 64×16 px range images for distances up to 20 m with a frame-rate of up to 100 Hz (cf. Fardi *et al.* [58]). Ground truth information is generated using a 2-D laser-scanner mounted on the car's radiator grille (see Fig. 3.2b).

As the proposed algorithm is designed to estimate translational motion, a large rubber ball is used due to its rotational invariance. The sequence (PMD) used for evaluation is shown in Fig. A.2 in appendix A. It is possible to reconstruct the ball's 3-D shape from the measured 2-D scanline, as both the ball's radius and the scanline's height are known. In the scene, the ball is pushed in front of the stationary car and – due to a slightly inclined ground plane – performs a curve to the left, heading back towards the car (cf. Fig. 4.13a).

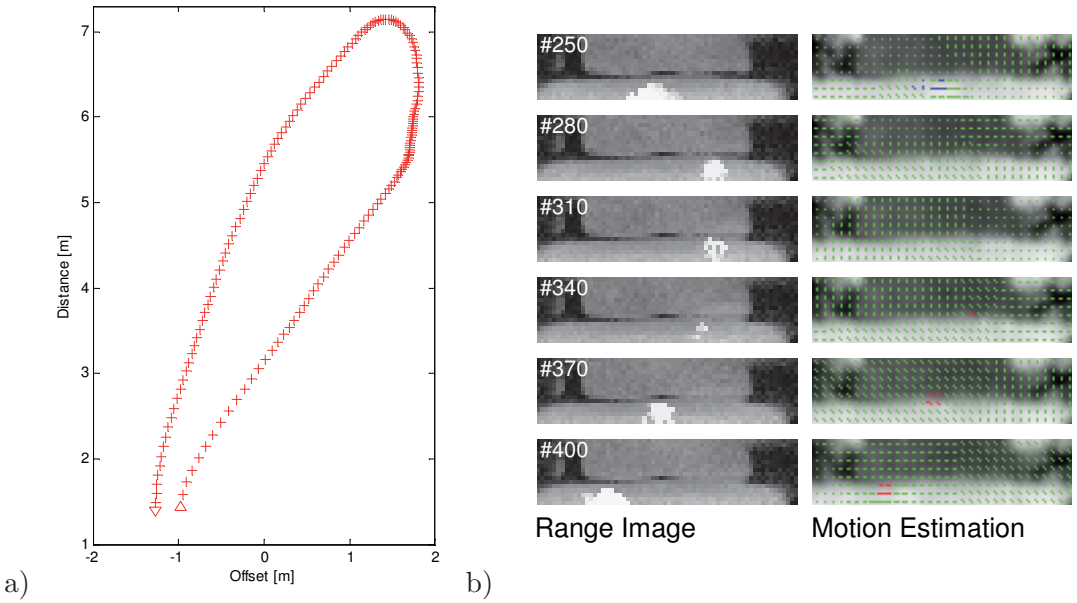
In order to determine the trajectory of the ball's centre, the readings of the laser-scanner are discarded unless they fall into a rectangle (distance 0 m to 10 m and offset -5m to 5m), which exclusively returns readings showing the ball. These readings fall onto a circle with the ball's radius. The ball's centre $(x_\odot, y_\odot)$ is determined fulfilling the circle equation Eq. 4.15 for the selected laser scanner readings (x_{LS}, y_{LS}).

$$x_\odot, y_\odot = \arg \ (x_{LS_{1,2,..,n}} - x_\odot)^2 + (y_{LS_{1,2,..,n}} - y_\odot)^2 \tag{4.15}$$

For $n > 2$, Eq. 4.15 is overdetermined, which is solved by averaging all centre positions which are calculated using 2 laser readings at a time. The centre positions are then processed by applying both median and Gaussian filters to get a continuous motion (see Fig. 4.13a).

A set of example frames from the range image sequence of the same scene acquired with a PMD device[4] is given in Fig. 4.13b. In order to be used with PCS, the range data

[4]Available online: http://www.matzka.net/vision/html/pmd.html The video shows the source range image, ground truth, motion estimation and motion vector field (from top to bottom).



a) b)

Figure 4.13: Scatterplot a) shows the ball's trajectory as detected with a laser scanner (Δ represents frame #250, ∇ frame #400). The range image sequence b) shows selected frames of the scene as seen by the PMD device (ball is brightened manually as to enhance visibility in the range image) as well as the corresponding estimated motion vector field. In the latter, blue arrows indicate an increasing distance, red arrows a decreasing distance.

is filtered over a small number of frames and outliers are rejected. Spatial filtering is not performed at this point, as the motion estimation algorithm includes this operation.

Generating the motion ground truth information from the laser readings is performed using the coordinate transformation function described by Eq. 4.4 to 4.6. The MSE values of the motion estimation for the acquired range image sequence as compared to ground truth are shown in Fig. 4.14.

Fig. 4.14 shows that both Gaussian filtering and median filtering of the motion vector field results in a considerable MSE reduction for both PCS, and FS. Due to the large fraction of (0,0,0) MVs in the ground truth, the FS is affected by incorrect MVs in the presence of unfiltered noise. Again, PCS performs significantly faster at 46.9 fps than FS with a framerate of 19.5 fps at a comparable motion vector quality.

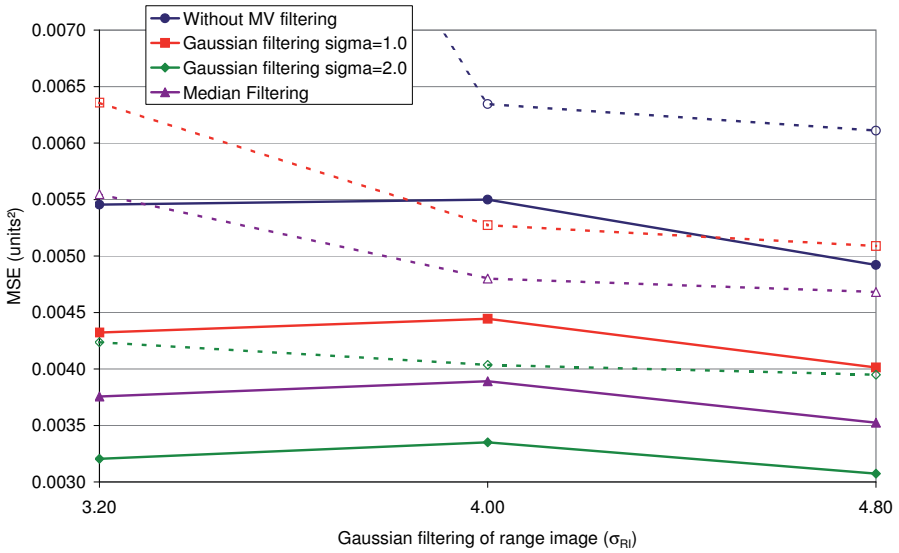


Figure 4.14: Mean squared error of motion vector components estimated by PCS_3 (solid line) and FS (dotted line) as compared to the ground truth under influence of Gaussian noise σ_{noise} for the orbiting movement pattern. The source range image is processed using a Gaussian filter with σ_{RI}.

4.6 Discussion of Data Level Modules

In this chapter, low-level data processing methods to convert sensor data towards a level of abstraction that are for data interpretation are presented. The used methods are computationally inexpensive, as they are continually performed on all acquired low-resolution sensor data. Apart from the level of abstraction this presents the main difference to the semantic level modules discussed in chapter 5, where only selected candidate regions are processed. Analogies between the human visual system (HVS) and the low-level image processing methods performed to obtain data level representation are pointed out in Tab. 4.4.

Property	Peripheral vision of HVS	Low-level image processing
Operation	continuous	continuous
Region	peripheral regions	entire image
Visual acuity	low acuity	low resolution
Features	luminance, range, motion	luminance, range, motion

Table 4.4: Analogy of the human visual system, and the low-level image processing methods performed to obtain data level representation.

Besides the analogies given in Tab. 4.4 it is argued in the literature that the peripheral vision of the HVS also detects salient regions, detects areas of similar movements, and even

identifies learnt forms and textures. However this is a question of where the boundary between data level and semantic level is drawn. In our proposed system, these processing steps are considered high-level data interpretation for two reasons. First, detecting learnt object shapes or salient regions require a data level representation such as luminance, range, or motion. Second, the results of detection processes are semantic information (e.g. traffic participant, salient region) as opposed to syntactic information (e.g. bright, near, slow) which is represented in our system's data level.

Chapter 5

Semantic Level

Semantic level representations are an abstraction of data level representations that can be obtained by performing high-level data interpretation methods. An interpretation of data level information is necessary in order to bridge the *semantic gap*, a term defined by Smeulders *et al.* [167] as

> *"the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation."* (Smeulders *et al.* [167])

In our proposed system the problem of transforming syntactical data in the data level towards semantic information that can be processed in a reasoning system is apparent. According to the definition given by Smeulders *et al.* [167] it is necessary to interpret data level information to bridge the semantic gap. In the following, each semantic representation is presented with respect to the available data and the required interpretation processes.

The road type concept used in our system is introduced in section 5.1. Traffic participant detection and classification using video data is discussed and evaluated in section 5.2, whereas the use of 3-D range information for traffic participant detection is discussed in section 5.3. Unsupervised estimation of salient regions as an alternative indicator besides traffic participant detection is proposed and evaluated in section 5.4. The concept of time-to-collision is presented in 5.5. In section 5.6 a discussion of the methods presented in this chapter is given.

5.1 Road Type

In our system, information about the current road type is used to determine the probability and severity level of accidents with other traffic participants. With the ego-vehicle's state vector $\vec{\tau}$ provided, the global position (p_{lat}, p_{lon}) is used to obtain the road type (RT) using a digital road-map as prior knowledge. The road selection process itself is performed by an external module containing the digital road-map. An algorithm for road matching and selection is also proposed by El Najjar and Bonnifait [168, 169]. The road type query returns one of five road types RT_n defined in our ontology in Fig. 1.2.

RT_1 pedestrian zone	RT_2 traffic-calmed road
RT_3 urban road	RT_4 country road
RT_5 motorway	

and a corresponding speed limit v_{max} if available.

The inference from locality towards road type is only valid as long as the current context on a given road type is consistent with this road type's predominant context. Therefore road type is used here as a means to describe context rather than locality. In our proposed system this is taken into account by adapting the current road type considering the current velocity of the ego-vehicle. This feature is a strong indicator of the current road type context provided that the human driver adapts his or her driving towards the current situation.

As an example, an urban road RT_3 determined using locality is altered into a traffic-calmed road type RT_2 information if the ego-vehicle's velocity does not exceed $v = 10\frac{m}{s}$. Slow-moving traffic allows pedestrians to cross the road or bicyclists to pass through car traffic, which is typical in a RT_2 context. For an ego-vehicle exceeding $v = 30\frac{m}{s}$, the same concept alters the road type context to RT_4 in accordance with a multi-lane situation devoid of pedestrians or bicyclists.

This example is generalised as follows. Every road type has a typical associated ego-vehicle velocity range of $v = [0\frac{m}{s},5\frac{m}{s}]$ for RT_1 towards $v > 30\frac{m}{s}$ for RT_5. The road type index n is then increased by one for every two velocity ranges it exceeds the typical associated velocity range. The same applies analogously if the velocity is lower than the typical associated ego-vehicle velocity. The resulting set of adapted road type indices n dependent upon locality and velocity is given Tab. 5.1. If the road type cannot be

determined using locality ($RT_?$), the typical associated velocity is directly mapped onto the adapted road type description.

	RT_n defined by locality					
$v[\frac{m}{s}]$	RT_1	RT_2	RT_3	RT_4	RT_5	$RT_?$
0..5	1	1	2	2	3	1
5..10	1	2	2	3	3	2
10..20	(2)	2	3	3	4	3
20..30	(2)	(3)	3	4	4	4
>30	(3)	(3)	(4)	4	5	5

Table 5.1: Road type indices n for road types RT_n defined by locality and adapted using the ego-vehicle's current velocity v. Indices written in parentheses are problematic due to the substantial violation of traffic rules, complicating a suitable categorisation of the current context with the given road type categories. If the road type is unknown $RT_?$, the ego-vehicle's velocity alone determines the adapted road type context.

5.2 2-D Traffic Participant Detection and Classification

In our presented system, luminance information acquired by video cameras is used to detect and classify traffic participants. For this, a set of trained classifier cascades as proposed by Viola and Jones [52] is used for both detection and classification. In our system, a distinction between object detection and object classification is made.

- Object *detection* is defined as the detection of a predefined category of objects, in our case traffic-participants (TP).

- Object *classification* is a refinement of object detection and is able to distinguish between different types of traffic participants TP_n,

using the five traffic participant types defined in our ontology in Fig. 1.2.

TP_1 pedestrian TP_2 bicycle

TP_3 motorcycle TP_4 car

TP_5 lorry

Below, the training process and performance evaluation of the detector and classifier cascades used in our system is discussed. For this both the role and the selection of negative samples is pointed out in section 5.2.1. Our classifier cascades are presented and evaluated in sections 5.2.2 to 5.2.4, followed by our detector cascades in section 5.2.5. Validation of detected traffic participants to decrease the number of false detections is described in section 5.2.6.

5.2.1 Training and Evaluation of Cascades

In the following, the selection of samples used for cascade training and the method for training and evaluation of detector cascades and classifier cascades[1] are discussed.

Samples for Cascade Training

A large pool of both positive and negative samples is used for cascade training. Negative samples do not contain any traffic participants for detector cascades. For classifier cascades, negative samples do not contain the traffic participant class used as positive samples for classifier cascade but contain traffic participants of a different class. This concept is also illustrated in Fig. 5.1.

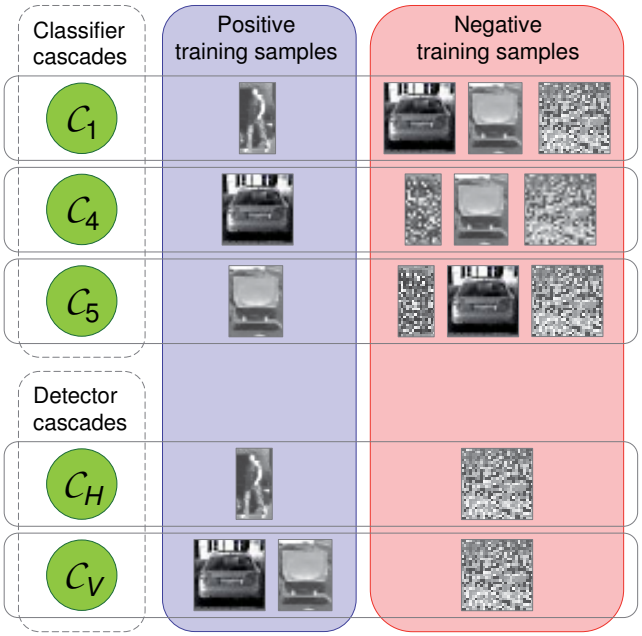


Figure 5.1: Training of the traffic participant detection and classification cascades. For each type of training samples a representative icon (pedestrian, car, lorry, and background) is used to represent which types are used for the training of which cascades.

Positive samples are obtained by manually labelling traffic participants in road traffic sequences, as well as from existing databases. In order to increase the number of positive samples, the samples are mirrored along their j-axis. This method results in only partly independent positive samples, but is a common method used in the literature (e.g. Munder and Gavrila [170]).

The number of background images without traffic participants used for negative sample

[1] All trained classifier cascades are available online: http://www.matzka.net/vision/html/cascades.html

generation is 350 images with a size of 640×480 px. From each background image, a large number of negative samples is generated by cropping and rescaling samples at different positions and scales. Using a background image of 640×480 px, a minimum sample size of 32×32 px and allowing position and scale steps of 2 px a total of

$$\sum_{i=j=32,34,\ldots,480} \left(\frac{(640-i)}{2} \cdot \frac{(480-j)}{2} \right) \approx 5.75 \cdot 10^6 \tag{5.1}$$

negative samples are generated from every background image. For 350 background images, this results in an overall number of $\approx 2 \cdot 10^8$ negative samples.

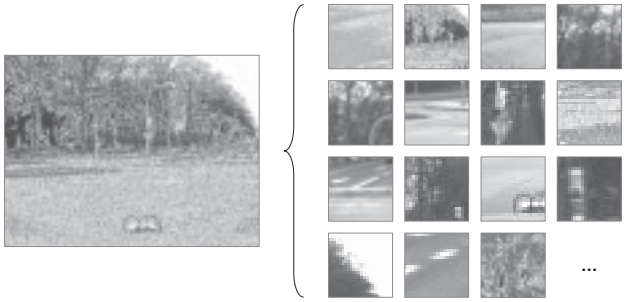


Figure 5.2: Generation of negative samples used for feature training. Negative samples are cropped and resized from a single example image towards a common sample size.

Training and Evaluation of Cascades

All cascades are trained using the *haartraining* tool of Intel's OpenCV image processing library[2] (cf. Bradski and Kaehler [53]). Every set of positive and negative samples is split into a training set containing 80% and a test set containing 20% of the total samples. For every stage, the cascade's overall performance on the test set is evaluated using the cascade's true positive rate $P(\mathcal{C}|\mathcal{TP})$ and false positive rate $P(\mathcal{C}|\neg\mathcal{TP})$. Every pair of measurements is represented by a single point in the classifier performance graphs (e.g. Fig. 5.2 for pedestrian classification). All data points of a cascade are then connected to show the performance of the training process.

5.2.2 Pedestrian Classifier Cascades

For pedestrian classifier training 750 positive samples with a resolution of 20×40 px are used. The positive samples are manually labelled from the video sequences acquired

[2] Available online at http://sourceforge.net/projects/opencvlibrary/

with the test-vehicle and supplemented with positive samples from the DaimlerChrysler Pedestrian Classification Benchmark Dataset[3] presented by Munder and Gavrila [170]. Each sample is mirrored along its j-axis to double the number of samples to a total of 1500 pedestrian samples. A subset of the used positive pedestrian samples is shown in Fig. 5.3.

Figure 5.3: Positive pedestrian samples with a resolution of 20×40 px manually selected from the video sequences acquired with the test-vehicle and supplemented with positive samples from the DaimlerChrysler Pedestrian Classification Benchmark Dataset.

Three cascades for pedestrian classification are trained with a minimum true positive rate of 0.997, 0.990, and 0.980 per stage for the first three stages. Beginning with the fourth stage, all cascades use a minimum true positive rate of 0.997 per stage. The Haar-like features of stages 0 and 1 of the pedestrian classifier cascade with a minimum true positive rate of 0.997 can be seen in Fig. 5.4.

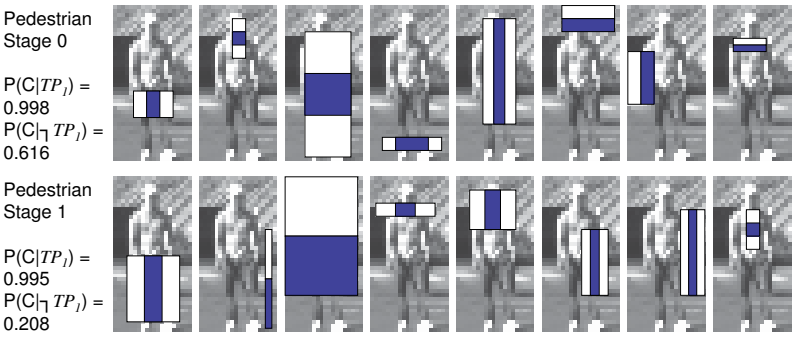


Figure 5.4: Haar-like features of stages 0 and 1 of the pedestrian classifier cascade with a minimum true positive rate of 0.997 shown in front of an example pedestrian image used for feature training. For each stage, the resulting rate of true positives $\mathrm{P}(\mathrm{C}|\mathcal{TP})$ and false positives $\mathrm{P}(\mathrm{C}|\neg\mathcal{TP})$ is given.

[3]The *DaimlerChrysler Pedestrian Classification Benchmark Dataset* is available online: http://www.science.uva.nl/research/isla/downloads/pedestrians/.

Evaluation of Pedestrian Classifier

Tab. 5.2 provides an overview of performance and computational costs of the trained pedestrian classifier cascades.

$P(C\|\mathcal{TP}_1)$ per stage		Cascade performance		Number of features	
Stages 0-2	Stages 3-29	$P(C\|\mathcal{TP}_1)$	$P(C\|\neg\mathcal{TP}_1)$	in cascade	mean per sample
0.997	0.997	0.8973	$9.10{\cdot}10^{-6}$	987	19.9
0.990	0.997	0.8919	$8.85{\cdot}10^{-6}$	664	15.5
0.980	0.997	0.8827	$7.99{\cdot}10^{-6}$	725	11.0

Table 5.2: Classifier performance and computational costs for three pedestrian classifier cascades with 30 trained stages and a minimum initial true positive rate of 0.997, 0.990, and 0.980.

It can be seen in Tab. 5.2 that the three pedestrian classifier cascades do not differ much in their performance. The true positive rates are within 1.5% and the false positives rates are approximatelly $8.5{\cdot}10^{-6}$ for all cascades. The computational costs differ significantly, as the mean numbers of features applied per sample are 11.0, 15.5, and 19.9 respectively. The overall number of features is high for all three cascades, yet the use of 987 features for the cascade with a minimum positive rate of 0.997 in particular suggest an overfitting as described in section 2.2.1.

Classifier Performance In order to assess the classification performance in detail, the true positive classification rates $P(C|\mathcal{TP}_1)$ and false positive classification rates $P(C|\neg\mathcal{TP}_1)$ at every stage are determined using a test set. The resulting graphs are drawn in Fig. 5.5.

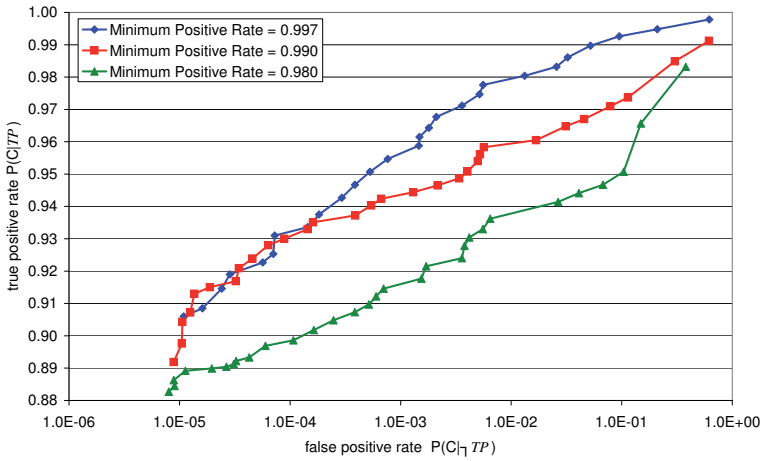


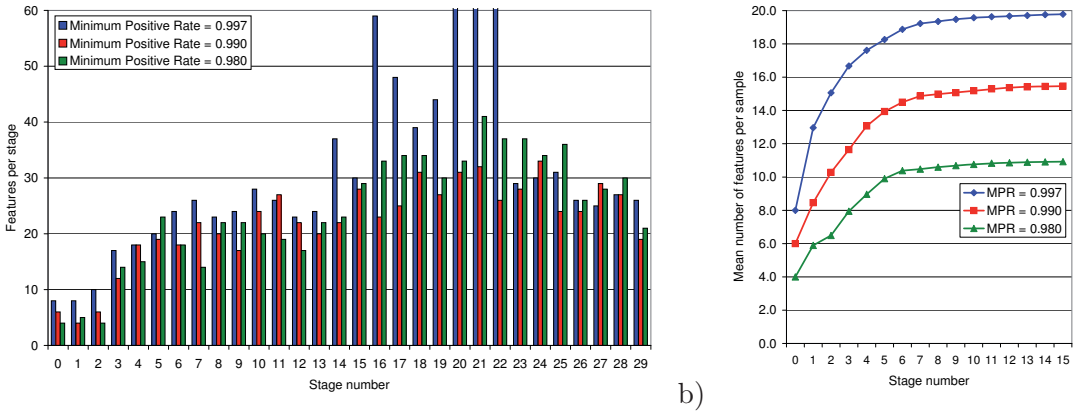
Figure 5.5: Classifier performance for three pedestrian classifier cascades with 30 trained stages and a minimum initial true positive rate of 0.997 (blue), 0.990 (red), and 0.980 (green). The false positive classification rate $P(C|\neg\mathcal{TP}_1)$ is drawn on a logarithmic scale.

The classification performance of the three pedestrian classifier cascades in Fig. 5.5 is far from the theoretical false positive rate after 30 cascades as

$$0.5^{30} = 9.3 \cdot 10^{-10} \ll 8.5 \cdot 10^{-6}$$

The classifier performance graphs in Fig. 5.5 for 0.997 (blue) and 0.990 (red) intersect, which is a second indicator for a non-ideal training process. The mean reduction of false positives per stage for the test set is approximately 0.3 where it is 0.5 for the training set. This lack of generality again indicates an apparent overfitting. One reason for this is that pedestrians samples are given in all poses and from all directions. This intra-class variability in appearance over all positive samples decreases the classification performance significantly.

Computational Costs The number of features at every stage allows the determination of the mean number of features applied per examined sample and substantiation of the claim that the pedestrian cascades are overfitted. Fig. 5.6a shows the number of features at every stage for the three pedestrian classifier cascades, in Fig. 5.6b the mean number of features used per sample is given.



a) b)

Figure 5.6: Number of features a) and mean number of features per sample b) at every stage for three pedestrian classifier cascades with 25 trained stages and a minimum initial true positive rate of $P(C|\mathcal{IP}_1)$=0.997 (blue), $P(C|\mathcal{IP}_1)$=0.990 (red), and $P(C|\mathcal{IP}_1)$=0.980 (green). Number of features at stages 20 to 22 exceed the value range of the diagram and are 82, 100, and 75 respectively.

Besides the large overall number of features per stage shown in Fig. 5.6a, the number of features for a minimum positive rate of 0.997 at stages 20 to 22 exceed the value range of the diagram and are 82, 100, and 75 respectively. These numbers in particular must be

seen as a strong indication for an overfitted cascade.

5.2.3 Car Classifier Cascades

For car classifier training 450 positive samples with a resolution of 32×32 px are manually selected from the video sequences acquired with the test-vehicle. Each sample is mirrored along its j-axis to double the number of samples to a total of 900 car samples. A subset of the used positive car samples is shown in Fig. 5.7.

Figure 5.7: Positive car samples with a resolution of 32×32 px manually selected from the video sequences acquired with the test-vehicle.

Three cascades for car classification are trained with a minimum true positive rate of 0.997, 0.990, and 0.980 per stage for the first three stages. Beginning with the fourth stage, all cascades use a minimum true positive rate of 0.997 per stage. The Haar-like features of stages 0 and 1 of the car classifier cascade with a minimum true positive rate of 0.997 can be seen in Fig. 5.8.

Figure 5.8: Haar-like features of stages 0 and 1 of the car classifier cascade with a minimum true positive rate of 0.997 shown in front of an example car image used for feature training. For each stage, the resulting rate of true positives $P(C|\mathcal{I}P_4)$ and false positives $P(C|\neg\mathcal{I}P_4)$ is given.

Evaluation of Car Classifier

Tab. 5.3 provides an overview of performance and computational costs of the trained car classifier cascades.

$P(C\|\mathcal{TP}_4)$ per stage		Cascade performance		Number of features	
Stages 0-2	Stages 3-24	$P(C\|\mathcal{TP}_4)$	$P(C\|\neg\mathcal{TP}_4)$	in cascade	mean per sample
0.997	0.997	0.9257	$6.54{\cdot}10^{-7}$	279	13.5
0.990	0.997	0.9180	$5.22{\cdot}10^{-7}$	280	11.1
0.980	0.997	0.9069	$1.08{\cdot}10^{-7}$	280	10.3

Table 5.3: Classifier performance and computational costs for three car classifier cascades with 25 trained stages of 0.997, 0.990, and 0.980.

The three car classifier cascades' performances in Tab. 5.3 are within a narrow range. The true positive rates are around 0.915, the false positives rates range from $1.08{\cdot}10^{-7}$ to $6.54{\cdot}10^{-7}$. The computational costs are also within a small margin of 10.3 to 13.5 features per sample on average. The overall number of features for all three cascades is low with 279 and 280 features.

Classifier Performance The true positive classification rates $P(C|\mathcal{TP}_4)$ and false positive classification rates $P(C|\neg\mathcal{TP}_4)$ at every stage determined using a test set are used to evaluate the car classifiers' performances. The resulting graphs are drawn in Fig. 5.9.

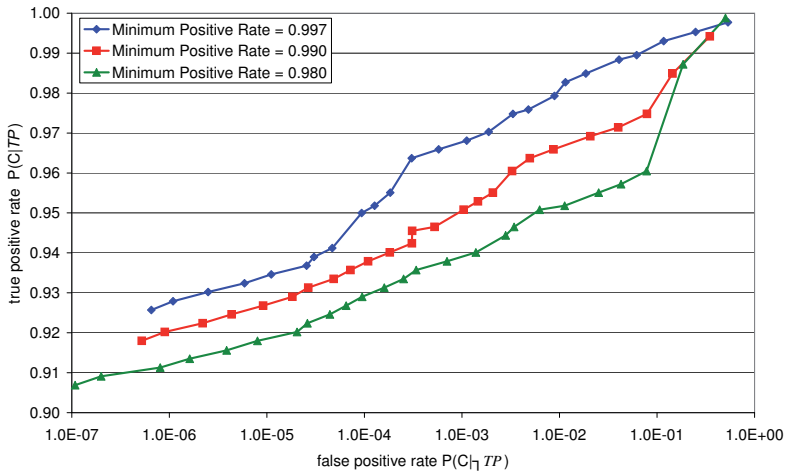


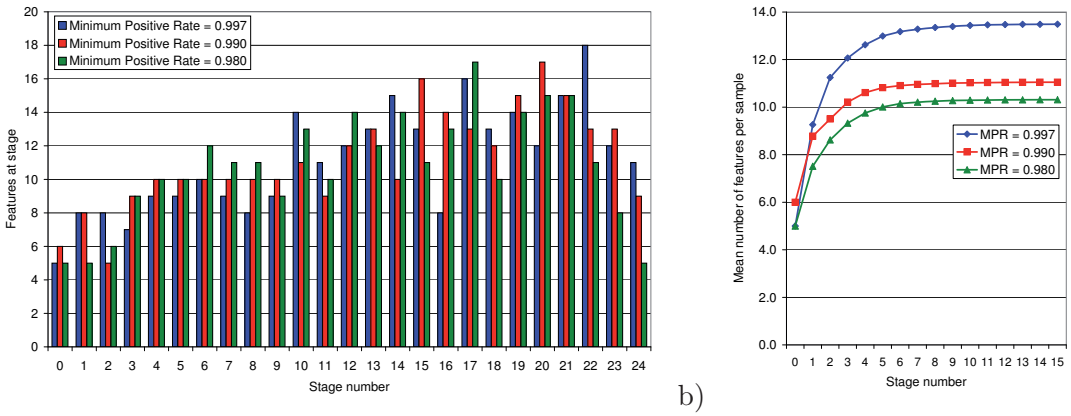
Figure 5.9: Classifier performance for three car classifier cascades with 25 trained stages and a minimum initial true positive rate of 0.997 (blue), 0.990 (red), and 0.980 (green). The false positive classification rate $P(C|\neg\mathcal{TP}_4)$ is drawn on a logarithmic scale.

The classification performance of the three car classifier cascades in Fig. 5.9 show a near-ideal decrease of false positives on the test set, indicating a good generalisation. The theoretical false positive rate of

$$0.5^{25} = 2.9 \cdot 10^{-8}$$

is also comparable to the measured false positive rates of $1.08{\cdot}10^{-7}$ to $6.54{\cdot}10^{-7}$ for our trained cascades.

Computational Costs As compared to the overall feature numbers for pedestrian classifier cascades, the car classifier cascades use approximately a third of the features (cf. Tab. 5.3) reducing the possibility of an overfitted classifier. The mean number of features used on each sample is small and ranges from 10.3 to 13.5. In Fig. 5.10a the number of features for all car classifier cascades at all stages is shown, as well as the mean number of features used per sample in Fig. 5.10b.



a) b)

Figure 5.10: Number of features a) and mean number of features per sample b) at every stage for three car classifier cascades with 25 trained stages and a minimum initial true positive rate of 0.997 (blue), 0.990 (red), and 0.980 (green).

5.2.4 Lorry Classifier Cascades

For lorry classifier training 70 positive samples with a resolution of 24×32 px are manually selected from the video sequences acquired with the test-vehicle. Each sample is mirrored along its j-axis to double the number of samples to a total of 140 car samples. This number of positive samples is far less than the 10^3 positive samples shown to perform a generalisable cascade training in the literature. To further increase the number of samples, each sample is rotated by 3° both clockwise and counter clockwise. This increases the number of positive samples to 420. A subset of the used positive lorry samples is shown in Fig. 5.11.

Three cascades for lorry classification are trained with a minimum true positive rate of 0.997, 0.990, and 0.980 per stage for the first four stages. Beginning with the fifth stage,

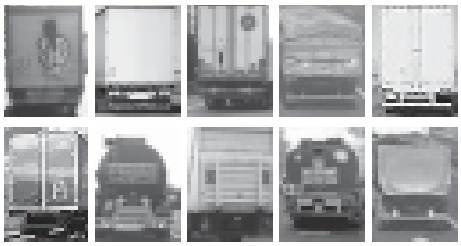


Figure 5.11: Positive lorry samples with a resolution of 24×32 px manually selected from the video sequences acquired with the test-vehicle.

all cascades use a minimum true positive rate of 0.997 per stage. The Haar-like features of stages 0 and 1 of the lorry classifier cascade with a minimum true positive rate of 0.997 can be seen in Fig. 5.12.

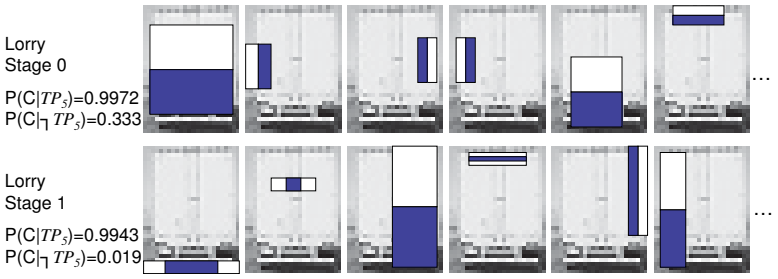


Figure 5.12: Haar-like features of stages 0 and 1 of the lorry classifier cascade with a minimum true positive rate of 0.997 shown in front of an example car image used for feature training. For each stage, the resulting rate of true positives $P(C|\mathcal{TP}_5)$ and false positives $P(C|\neg\mathcal{TP}_5)$ is given.

Evaluation of Lorry Classifier

Tab. 5.4 provides an overview of performance and computational costs of the trained lorry classifier cascades.

$P(C\|\mathcal{TP}_5)$ per stage		Cascade performance		Number of features	
Stages 0-3	Stages 4-21	$P(C\|\mathcal{TP}_5)$	$P(C\|\neg\mathcal{TP}_5)$	in cascade	mean per sample
0.997	0.997	0.9485	$6.22{\cdot}10^{-7}$	228	12.1
0.990	0.997	0.9229	$2.60{\cdot}10^{-7}$	238	10.0
0.980	0.997	0.9061	$4.97{\cdot}10^{-7}$	226	9.2

Table 5.4: Classifier performance and computational costs for three lorry classifier cascades with 22 trained stages and a minimum initial true positive rate of 0.997, 0.990, and 0.980.

Classifier Performance The true positive classification rates $P(C|\mathcal{TP}_5)$ and false positive classification rates $P(C|\neg\mathcal{TP}_5)$ at every stage determined using a test set are used to

evaluate the lorry classifiers' performances. The resulting graphs are drawn in Fig. 5.13.

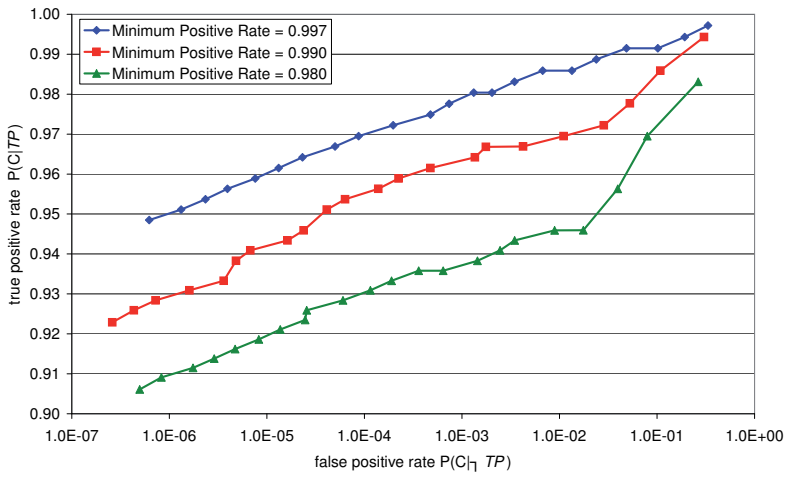


Figure 5.13: Number of features a) and mean number of features per sample b) at every stage for three lorry classifier cascades with 22 trained stages and a minimum initial true positive rate of 0.990 (blue). The false positive classification rate $P(C|\neg \mathcal{TP}_5)$ is drawn on a logarithmic scale.

The three lorry classifier cascades' performances in Fig. 5.13 show a near ideal training performance. The true positive rates range from 0.906 to 0.949 which can be expected considering four stages with different minimum positive rates. The false positives rates range from $2.60{\cdot}10^{-7}$ to $6.22{\cdot}10^{-7}$. Of all trained classifiers, these are closest to the theoretical false positive rate of

$$0.5^{22} = 2.38^{-7}$$

Computational Costs The computational costs of the lorry classifier cascades are low and within a small margin of 9.2 to 12.1 features per sample on average (cf. Fig. 5.14b). The overall number of features for all three cascades is low with 228 to 238 features. The features used per stage are displayed in Fig. 5.14a, the mean feature used per examined sample is given in Fig. 5.14b.

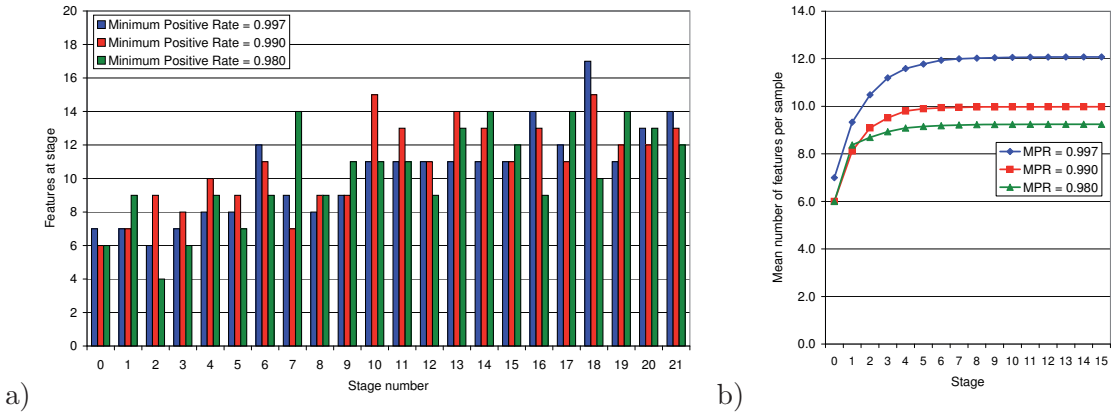


Figure 5.14: Number of features a) and mean number of features per sample b) at every stage for three lorry classifier cascades with 22 trained stages and a minimum initial true positive rate of 0.997 (blue), 0.990 (red), and 0.980 (green).

5.2.5 Human Detector Cascade and Vehicle Detector Cascades

In principle, traffic participant detection can be performed using a single detector cascade that discerns between traffic participant samples and background samples. In practice however, the differences between all possible traffic participant types are too significant. Therefore two detector cascades with smaller intra-class differences are trained: a human detector cascade to detect pedestrians (and eventually bicycles and light motorcycles) and a vehicle detector cascade to detect cars and lorries. Both cascades are trained with downscaled positive and negative samples to operate on low-resolution images.

For the human detector cascade 750 pedestrian samples are used for training. The positive samples are manually labelled from the video sequences acquired with our test-vehicle. Each sample is mirrored along its j-axis to double the number of samples to a total of 1500 human samples. The vehicle detector cascade is trained using 450 car samples and 70 lorry samples. Again, each sample is mirrored along its j-axis to double the number of samples to a total of 1040 vehicle samples.

Both cascades are trained with a minimum true positive rate of 0.990 per stage for the first three stages. Beginning with the fourth stage, the cascades use a minimum true positive rate of 0.997 per stage. The Haar-like features of stages 0 and 1 for both detector cascades can be seen in Fig. 5.15.

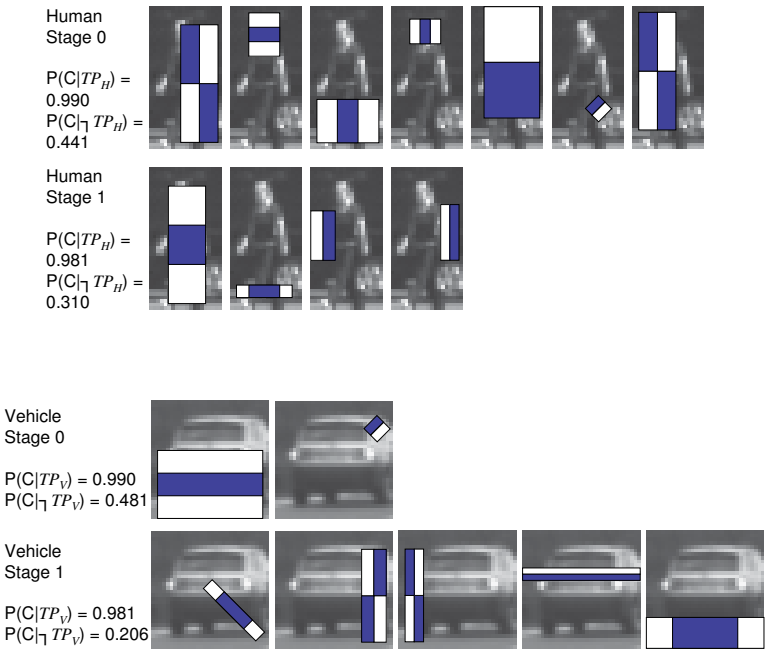


Figure 5.15: Haar-like features of stages 0 and 1 of the human detector cascade and vehicle detector cascade shown in front of an example image used for feature training. For each stage, the resulting rate of true positives P(C|$\mathcal{TP}$) and false positives P(C|$\neg\mathcal{TP}$) is given.

Evaluation of Detector Cascades

Tab. 5.5 provides an overview of performance and computational costs of the trained detector cascades.

	Detector performance		Number of features	
$P(C\|\mathcal{TP})$ per stage	$P(C\|\mathcal{TP})$	$P(C\|\neg\mathcal{TP})$	in cascade	mean per sample
Human Detector Cascade $\mathcal{C}_H$	0.9074	$6.75{\cdot}10^{-6}$	385	16.7
Vehicle Detector Cascade $\mathcal{C}_V$	0.9193	$1.41{\cdot}10^{-5}$	132	7.4

Table 5.5: Detection performance and computational costs for a human detector cascade with 27 stages and a vehicle detector cascade with 23 stages.

It can be seen in Tab. 5.5 that the human detector cascade has a lower true positive detection rate $P(C|\mathcal{TP})$ caused by four more stages in the human detector cascade. The false positive detection rates are comparable with approximately 10^{-5}. A major difference can be seen in the number of features both in the cascade and the mean features used per sample.

Detector Performance The true positive classification rates $P(C|\mathcal{TP})$ and false positive classification rates $P(C|\neg\mathcal{TP})$ at every stage determined using a test set are used to evaluate the detector cascades' performances. The resulting graphs are drawn in Fig. 5.16.

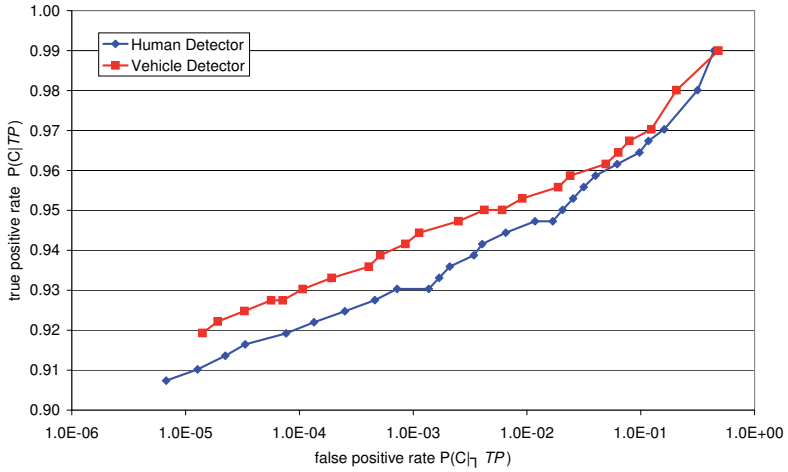


Figure 5.16: Detector performance for detector cascades with 27 trained stages for the human detector cascade (blue) and 23 trained stages for the vehicle detector cascade (red). The false positive classification rate $P(C|\neg\mathcal{TP})$ is drawn on a logarithmic scale.

The detection performance for both detector cascades in Fig. 5.9, but in particular the human detector cascade, show a non-ideal decrease of false positives which also substantiates in the difference between the measured overall false positive rates and the theoretical false positive rates after 23 and 27 stages

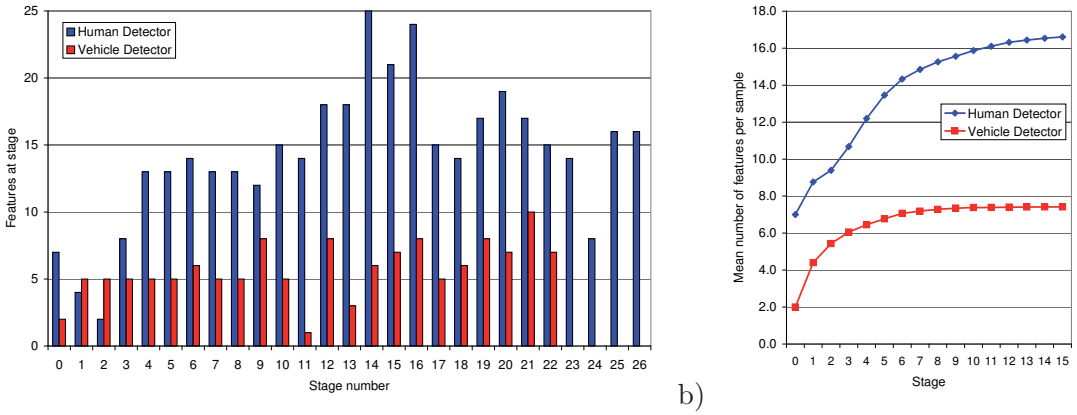
$$0.5^{23} = 1.2 \cdot 10^{-7}, \quad 0.5^{27} = 7.5 \cdot 10^{-9}$$

considering measured false positive rates of $6.75{\cdot}10^{-6}$ for the human detector cascade and $1.41{\cdot}10^{-5}$ for the vehicle detector cascade.

Computational Costs In Fig. 5.17 the number of features for both detector cascades at all stages and the mean number of features used on each sample is shown.

Frequency of False Positives

For an image of 640×480 px and a minimum sample size of 32×32 px, a total of $5.75{\cdot}10^6$ samples must be examined for every video frame (cf. Eq. 5.1). Using a downsampled video frame of 320×240 px and a minimum sample size of 16×16 px, reduces the total number of samples to $7.3{\cdot}10^5$.



a) b)

Figure 5.17: Number of features a) and mean number of features per sample b) at every stage for the human detector cascade (blue) and the vehicle detector cascade (red).

Considering an average false positives rate of 10^{-5} for both detector cascades, the theoretical number of false positives $N_{(C|\neg\mathcal{TP})}$ assuming an equal distribution over the whole image amounts to

$$N_{(C|\neg\mathcal{TP})} = (10^{-5} + 10^{-5}) \cdot 7.3 \cdot 10^5 = 14.6 \tag{5.2}$$

per video frame using two detection cascades. This large number of false positives requires a validation of detected traffic participants, which is described below.

5.2.6 Validation of detected Traffic Participants

The number of false positives in every frame necessitate a validation of all detected traffic participants. This validation is performed by cropping the image, checking the region's size and position, and by removing nested detections.

Rules used for Validation

First, part of the video frame covers the test vehicle's bonnet. Detected traffic participants inside this area or with a significant overlap with the bonnet are discarded. It is computationally effective to discard this area prior to applying the detection cascades. For our sequences this reduces the low-resolution image to 320×200 px which in turn reduces the regions which must be examined for every video frame to $5.2{\cdot}10^5$ assuming a minimum sample size of 16×16 px (cf. Eq. 5.1 and section 5.2.5). Compared to $7.3{\cdot}10^5$ for a 320×240 px pixel image this constitutes a reduction of 28.8% for computation time.

The reduced number of processed samples however shows only a minimal impact on the number of false positives, as these rarely appear on the vehicle's bonnet, the latter being included in the negative samples of the training set.

Second, the correlation between the size and position of a traffic participant inside the image is used. Using ground truth information, the height and the bottom y-coordinate of every traffic participant's bounding box is obtained. A scatter plot of the former is given in Fig. 5.18.

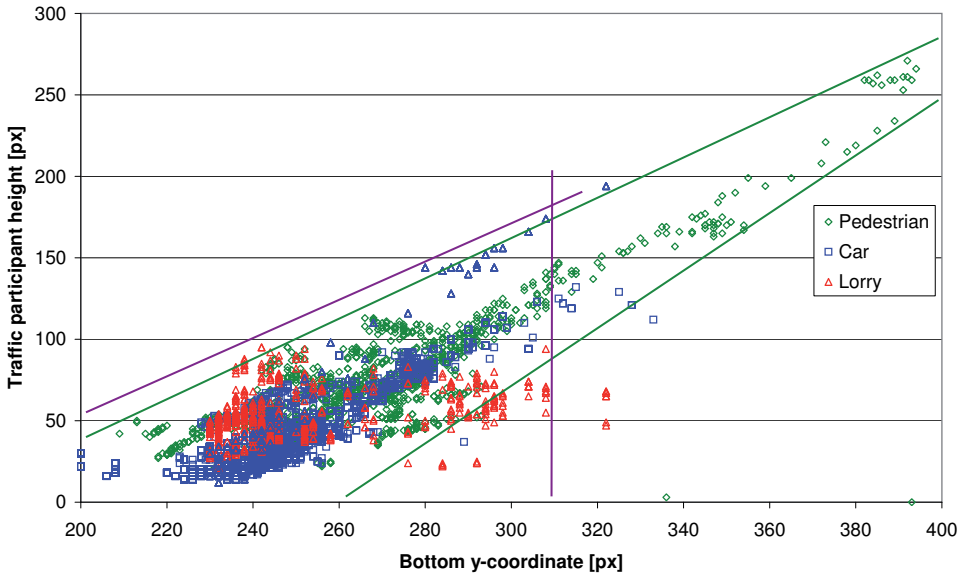


Figure 5.18: Scatter plot showing the correlation between the height and the bottom y-coordinate of every traffic participant's bounding box for pedestrians (green), cars (blue), and lorries (red). Established constraints for both detected human traffic participants (green) and vehicles (violet) are shown as coloured lines.

The scatter plot in Fig. 5.18 shows that a strong correlation between the height and the bottom y-coordinate of a bounding box exists. This correlation is used to establish two constraints for both detected human traffic participants (green, Eq. 5.3) and vehicles (violet, Eq. 5.4) shown as coloured lines in Fig. 5.18 that contain 98% of all bounding boxes. To be considered valid, the detected bounding box must fall into the area enclosed in the constraints.

$$\mathcal{TP}_H = \begin{cases} invalid & \text{if } j_{UL} - j_{LR} > 1.029 \cdot j_{LR} - 159.2, \\ invalid & \text{if } j_{UL} - j_{LR} < 0.536 \cdot j_{LR} - 105.1, \\ valid & \text{otherwise} \end{cases} \tag{5.3}$$

$$\mathcal{TP}_V = \begin{cases} invalid & \text{if } j_{UL} - j_{LR} > 0.840 \cdot j_{LR} - 105.0, \\ invalid & \text{if } j_{LR} > 310, \\ valid & \text{otherwise} \end{cases} \tag{5.4}$$

As opposed to discarding detected regions not meeting the constraints given in Eq. 5.3 and 5.3 the use of a smart sliding window concept for the Viola and Jones classifier cascade can be considered to increase efficiency. However, the presented concept of discarding invalid regions is preferable in the implemented system due to the complexity of the OpenCV library used.

Third, a common case for a false positive is that of nested detection, where the same traffic participant is detected at different scales. A detection is considered to be nested if the centre of the smaller region is located inside a larger region. In this case, only the largest detected traffic participant is maintained while all smaller detections are discarded. If the nested traffic participants are from different detectors, the type of the maintained, largest detected traffic participant is used.

Evaluation of Validation

In order to evaluate the validation process performed on detected traffic participants, the number of false positives after detection discarding the vehicle's bonnet, after validation using the detected traffic participants' heights and positions, and after removing nested detections is measured. For this, the test sequences with a total of 1512 video frames including 2974 traffic participants are used. The mean number of false positives per frame for both our human traffic participant detector (green) and our vehicle detector (violet) are given in Fig. 5.19.

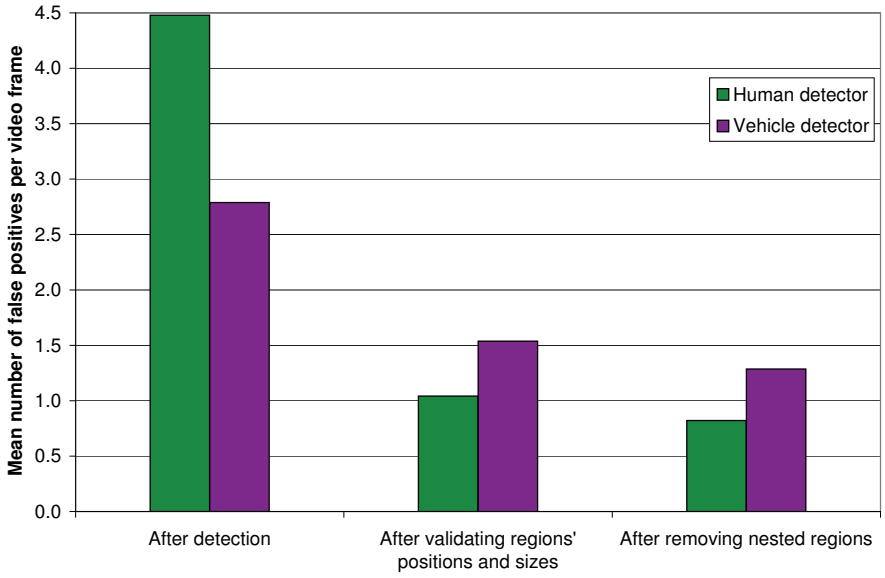


Figure 5.19: Mean number of false positives after detection discarding the vehicle's bonnet, after validation using the detected traffic participant's height and position, and after removing nested detections for our human traffic participant detector (green) and our vehicle detector (violet).

In Fig. 5.19 a reduction of the mean number of false positives per frame from a total of $P(C|\neg\mathcal{IP}) = P(C|\neg\mathcal{IP}_H) + P(C|\neg\mathcal{IP}_V) = 7.3$ false positives before validation to $P(C|\neg\mathcal{IP}) = 2.6$ false positives per frame using the correlation between the position and height of the bounding box is shown. A further reduction to $P(C|\neg\mathcal{IP}) = 2.1$ false positives per frame is achieved after removing nested detections. It can also be seen that the mean number of false positives using our human traffic participant detector is considerable with 4.5 false positives per frame, but is also significantly reduced by the validation. For our vehicle detector, the number of false positives is smaller after detection, but is not reduced as much as the false human traffic participant detections, due to a higher variance in size considering both cars and lorries.

5.3 3-D Traffic Participant Classification

Our research included the implementation and evaluation of a 3-D traffic participant classification concept based upon spin images (cf. section 2.2.2). For this, a controllable laser scanner is assumed to be available for our sensor system.

The implementation and evaluation of a method to determine efficient scan-patterns to acquire a sparse spin image representation is first proposed in Matzka *et al.* [14]. The

classification results using spin images are weighted against the cost associated with introducing a controllable laser scanner in the test vehicle's sensor system. It is pointed out in the literature review, that sensors for driver assistance systems are primarily chosen considering the sensor's cost as opposed to the sensor quality requirement for autonomous driving systems.

From the evaluation of the attainable classification of our prototype implementation, a substantial gain in classification quality as opposed to 2-D classification using video images could not be shown. At the same time, using either a 3-D laser scanner or a controllable laser scanner is expensive. Therefore the use of a controllable laser scanner on our test vehicle and thus in our presented system, is dismissed. However, considering the use of 3-D laser scanners in future systems, the spin image generation using sparse input data is described in the following.

5.3.1 Spin Image Generation with sparse Input Data

Object classification relies heavily on an accurate knowledge about the car's environment. One way to gain this knowledge is the use of range sensors such as radars or laser scanners. The latter are often capable of acquiring high-resolution range-information, yet it is very time-consuming to obtain a regular set of input data. This would for example require the scene to be scanned line by line. In a dynamic road traffic environment this becomes problematic, as a single 3-D scan of the environment is reported to require 4 s to 12 s by Surmann *et al.*[41].

There exist measurement concepts other than obtaining a regular range information, such as by using Lissajous figures as proposed by Blais *et al.*[171] or by deflecting a 2-D scanline. Yet, even these concepts do not satisfy real-time constraints, if much data has to be acquired in order to perform a successful classification of a scanned object. It is necessary to generate an efficient scan pattern, that acquires sparse input data for a robust classification scheme, such as spin images. In this context, efficiency stands for an optimum cost-benefit-ratio which is the case if a good classification result can be obtained with few scanlines.

In Matzka *et al.*[14] we propose to obtain only a small number of scan-lines around an oriented point, which is depicted in Fig. 5.20.

This concept implies a large reduction of data, which is desirable. Moreover, the density of contributing points around the oriented point – and therefore on the object – is

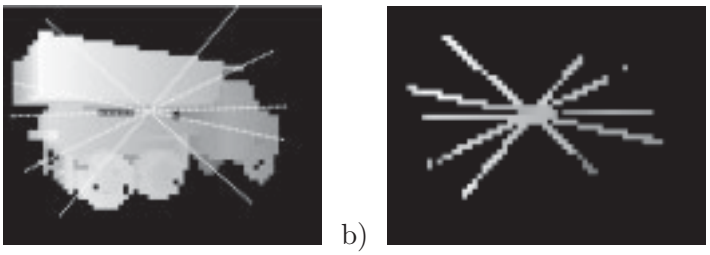


Figure 5.20: Acquisition of sparse data using five radial scanlines through an oriented point (a). For synthetic range-image (a), this has been emulated by masking out pixels not touched by a scanline (b) using the remaining pixels as input information.

relatively high, therefore the ratio of contributing points increases as compared to using a regular point-set.

Possible measurement concepts using scanlines with varying inclination angles can be both found in patents – mainly omni-directional bar-code scanners – and literature. Blais *et al.*[171] describe a triangulation based laser scanner using Lissajous scan patterns as opposed to obtaining a regular grid.

Another concept is to deflect the scanline of a 2-D time-of-flight laser scanner in a way so that the inclination angle is variable. This concept has been realised using two mirrors, which are independently rotated by two high-resolution stepper motors (see Fig. 5.21 below).

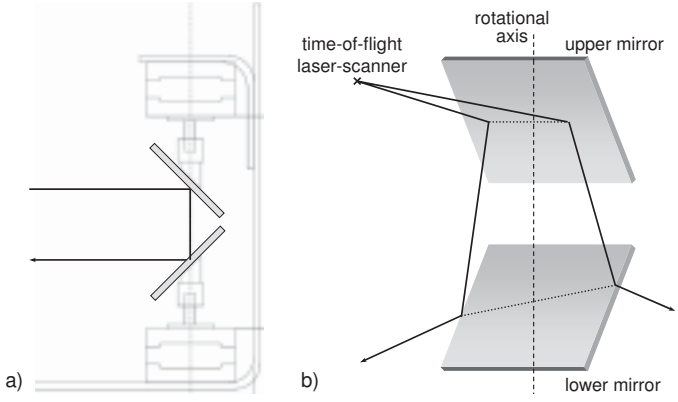


Figure 5.21: Figure a) shows a sketch of the deflection concept with the two mirrors rotated by two stepper motors. Figure b) depicts the deflection and inclination of an initially horizontal scanline by the two rotated mirrors.

A set of scanlines intersecting at the same oriented point is defined to be a scan pattern. In the following sections we investigate which scan patterns are most suitable for the scanning process considering a spin image classification concept. For this, we generate a large database of random scan patterns and use the scan patterns to classify objects

using a spin image classifier. A linear regression method then determines the correlation of scan pattern features and the correct classification rate.

Generation of Scan Pattern Database

In order to generate a database of scan patterns with corresponding classification performance measures, 6177 scan patterns consisting of random radial lines through an oriented point are used on a test-set of 40 range-images, containing five different objects of four different classes (cf. Fig. 5.22).

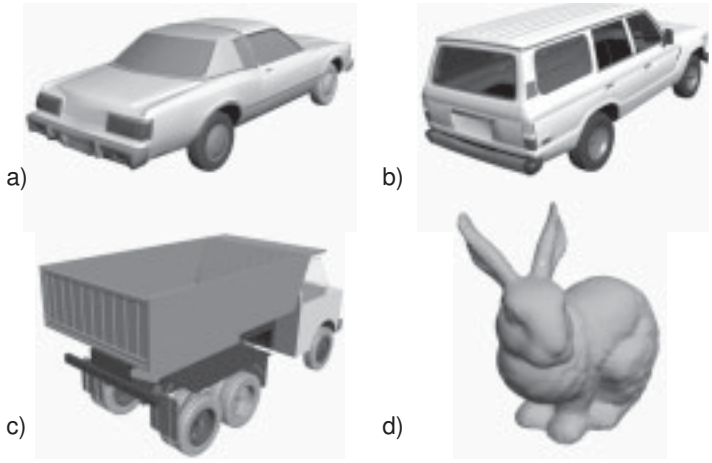


Figure 5.22: Four 3-D models used in the test-set, representing a) car, b) SUV, c) truck, and d) bunny. The fifth model (ND) is also a car and therefore belongs to the 'car' class

The models' eight range-images in the test-set are acquired from eight viewpoints by rotating each object around its z-axis, which is the only mayor rotational object motion, and therefore viewpoint-change, to be expected in road traffic scenes.

The classification performance is then determined by masking out pixels in a single range-image not touched by a scanline (cf. Fig. 5.20), using the remaining sparse range-data as input for the spin image classification algorithm.

The classification was then considered successful if the classification result was correct and the matching spin images from the test-set and the classification-database were acquired at the same oriented-point, which are determined using a geometrical saliency algorithm. The correspondence of oriented-points in the database and the test-set was defined manually beforehand, which could be done, as all scan patterns were applied at identical oriented-points.

Narrowing the definition of correct classification was necessary, as a correct classifi-

cation not based upon the correct oriented-point may well be considered an erroneous classification. This decreases the chance of correct classification by pure chance to 5% as opposed to 25% if the correct object class would suffice.

The classification rates range from 7.5% to 75.0%, with a median classification rate of 37.5% (cf. Fig. 5.23), distinguishing between 20 oriented-points.

5.3.2 Regression of Scan Pattern Features

Considering the distribution of classification rates shown in Fig. 5.23, the question arises, whether the classification rate is influenced by the chosen scan pattern, and if so, which features of the scan patterns show the highest correlation to the classification rate.

It is possible to use a multivariate, linear regression model with

$$y_i = \Theta_0 + x_{i1}\Theta_1 + \cdots + x_{ip}\Theta_p + e_i \quad (i = 1, \cdots, n) \tag{5.5}$$

with the error e_i exhibiting a normal distribution around zero.

A multivariate regression algorithm estimates the regression coefficients $\Theta = (\Theta_1, \cdot, \Theta_p)$ from a length i set of p predictor variable observations $x_{i1}, \cdots, x_{ip}$ and response variable observations y_i.

The most popular estimation technique for Θ is the sum of least squares method, where the sum of the squared residuals is minimised. However, this method is not robust against outliers, which can be measured by the notion of the breakdown point $\epsilon*$, which is the smallest percentage of outliers, that is able to cause the estimator to return an arbitrarily large deviant value (cf. Hampel [172] and Rousseeuw [173].

Using a sum of the least squares method, $\epsilon* = 0$, which is not robust at all. Besides

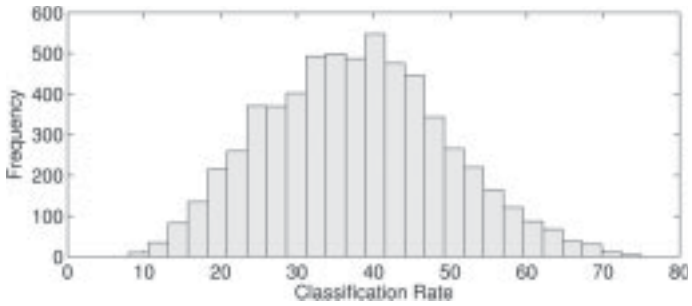


Figure 5.23: Distribution of classification rates distinguishing between 20 oriented-points in the database, ranging from 7.5% to 75.0%, with a median classification rate of 37.5%.

replacing the square with another functional it is possible to replace the sum of the squared errors by the median of the squared errors, called least median of squares (LMS, cf. Eq. 5.6).

$$\Theta_{1..p} = \arg\min(med\ r_i^2) \tag{5.6}$$

The LMS method is proven to posses $\epsilon* = 50\%$ as a breakdown point, but shows a slow convergence rate of $n^{-1/3}$. This problem can be mitigated by computing a one-step M estimator, which converges at $n^{-1/2}$ (cf. Bickel [174]). As a result, the least trimmed squares (LTS) method given by

$$\Theta_{1..p} = \arg\min(\sum_{i=1}^{h}(r^2)_{i:n} \tag{5.7}$$

where $(r^2)_{1:n} \leq \cdots \leq (r^2)_{n:n}$ are the ordered squared residuals [173]. This method allows a trimming proportion α' which determines the breakdown point $\epsilon*$ of the algorithm, as

$$\alpha' = \frac{1}{2} - \frac{p-1}{2\,n} \tag{5.8}$$

An implementation of the FAST-LTS method presented by Rousseeuw and Van Driessen [175] is included in the LIBRA library for MATLAB [176], and is used for the regression in our evaluation.

In order to determine the goodness of the fit of the regression, the unadjusted coefficient of determination R_u^2 (cf. Eq. 2.5) can be determined as

$$R_u^2 = \frac{cov(A,B)^2}{var(A) \cdot var(B)} \tag{5.9}$$

The problem with the unadjusted R_u^2 measure in Eq. 5.9 is that is will increase with the number of used coefficients, albeit slowly. This effect can be countered using the adjusted coefficient of determination R^2 according to Eq. 5.10, which is used throughout this chapter.

$$R^2 = 1 - (1 - R_u^2)\frac{n-1}{n-p-1} \tag{5.10}$$

Feature Selection

We select eight features x_i of scan patterns to be used in the multivariate regression method. The chosen features exhibit a correlation coefficient with the correct classification rate from 0.001 to 0.432, resulting in an overall correlation coefficient of $R^2 = 0.8$.

Number of Scanlines A major feature of the examined scan pattern is its number of scanlines. In the database, the latter ranges from 5 to 15, and shows a strong correlation with the measured classification rates (cf. Fig. 5.24).

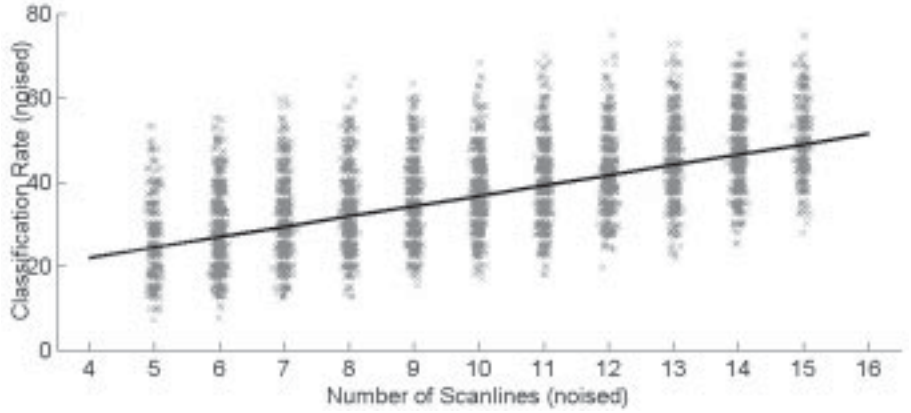


Figure 5.24: Correct classification rate drawn against number of scanlines. Note that both classification rate and number of scanlines have been slightly noised as they are both discrete values and would therefore hide the frequency of the individual value pairs. The black line shows the linear regression calculated using LTS regression.

It can be seen from Fig. 5.24, that the classification rate is correlated to the number of scanlines, and therefore the amount of range information, used. However, regarding $R^2 = 0.432$, there remains a considerable variance of the classification rate that is unaccounted for by the number of scanlines.

This implies that classification results can be optimised without increasing the number of scanlines and thus rendering the scanning process more efficient. It appears feasible to gain 40% classification rate with only five scanlines, which could only be expected to be the case using ten or more scanlines according to the regression.

The number of scanlines is the first predictor variable for the multivariate regression and will be referred to as x_{i1}.

Statistical Classification Performance of individual Inclination Angles The classification performance of an individual scanline's inclination angles can be assessed

by examining which classification rates have been achieved if a certain inclination angle has been used. The outcome of this examination can be seen in Fig. 5.25.

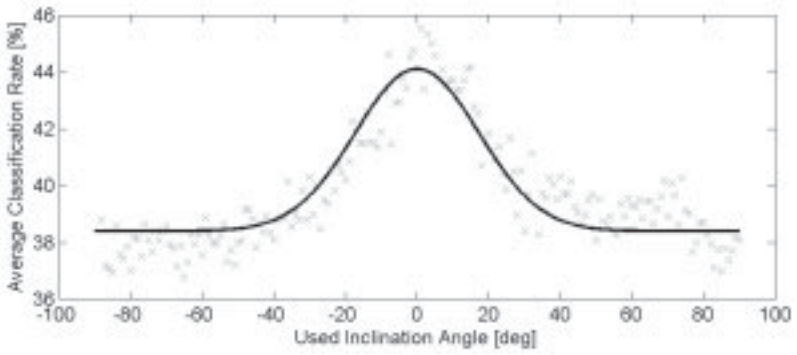


Figure 5.25: Average classification performance if the respective inclination angle has been used in the classification process. The black line shows a mean squared error approximation of the measurements with a Gaussian function.

The measurements in Fig. 5.25 can be approximated by a Gaussian function. For the average classification measurements in Fig. 5.25, a Gaussian approximation using a mean squared error method can be established (cf. Eq. 5.11)

$$y(x) = 246 \cdot \frac{e^{\left(\frac{-x^2}{2\cdot(17.2)^2}\right)}}{17.2 \cdot \sqrt{2\pi}} + 38.4 \tag{5.11}$$

A reason for the displayed behaviour is that the 3-D models are only rotated around their z-axis in order to obtain a various viewpoints, therefore the quality of the horizontal component of the surface normal is crucial to a correct classification, which is naturally improved by a horizontal scanline.

Neither the measurements in Fig. 5.25 nor their Gaussian approximation in Eq. 5.11 can be used for regression directly, as various inclination angles are used in order to acquire range-data. Out of the functions that have been tested, the maximum classification rate x_{i2} of all used inclination angles (R^2=0.337) and the average classification rate x_{i3} of all used inclination angles (R^2=0.329) show the highest correlation to the recognition rates gained by the corresponding scan patterns. However, the covariance between both values is 0.542, therefore we do not have two truly independent input variables.

Evenness of the scanlines' distribution Examining scan patterns that returned a high classification-rate, it appeared that in the majority of the cases, the scanlines were evenly distributed over the range of inclination angles. Robust linear regression resulted in

a R^2=0.229, showing a decreasing classification rate with an increasing standard deviation x_{i4} of the inclination angles' differences.

Fourth Central Moment (Kurtosis) of Inclination Angles During the search for statistical properties of the used scanlines that correlate with the classification rate, the fourth central moment of the used inclination angles x_{i5} – also called kurtosis (e.g. Joanes and Gill [177]) – has shown a coefficient of determination of R^2=0.107 with the classification rate measured.

Despite a high correlation towards the angle range (R^2=0.595) the fourth central moment improves the coefficient of determination if used as an additional predictor variable.

Angle range covered by the scanlines As expected, the angle range covered by the scanlines shows a correlation towards the classification rate. The angle range is defined to be 180° reduced by the largest distance between two angles. Performing LTS regression on the angle range versus the classification rate returned a R^2=0.019, which is comparatively small.

However, the angle range feature x_{i6} has a considerable statistical impact if it is combined with the number of scanlines. In a robust multivariate regression, it increases the number of scanlines' R^2_{i1}=0.432 to $R^2_{i[1;6]}$=0.552 if both input variables are considered. This is an emergent behaviour that is sometimes seen in multivariate regression.

Median Inclination Angle of used Scanlines The median inclination angle of the used scanlines x_{i7} shows a small correlation towards the classification rate with R^2=0.010. As this feature is largely independent from the other features, it is useful to include x_{i7} in the multivariate regression.

Number of Scanline-Clusters Besides the number of scanlines, the number of scanline-clusters x_{i7}, which was acquired using a k-means clustering algorithm, has shown to be of interest. This is not so much caused by the direct correlation, which is as little as R^2=0.001, but in connection with the number of scanlines and/or the evenness of the inclination angle's distribution. As opposed to R^2=0.450 when using only number of scanlines (x_{i1}) and evenness (x_{i4}) in a multivariate regression, a coefficient of determination of R^2=0.493 is gained with the number of scanline clusters as an additional prediction variable.

Multivariate Regression

Providing LIBRA's LTS regression algorithm with the eight predictor variables and the classification rate as the response variable, the regression coefficients $\Theta_{1..p}$, an offset Θ_0, and a coefficient of determination of $R^2 = 0.800$ (cf. Eq. 5.12 below) are determined.

$$y_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{i8} \end{pmatrix}' \cdot \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_8 \end{pmatrix} + \Theta_0 = \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ x_{i5} \\ x_{i6} \\ x_{i7} \\ x_{i8} \end{pmatrix}' \cdot \begin{pmatrix} 3.27 \\ 1.23 \\ 8.64 \\ 0.27 \\ 18.14 \\ -0.22 \\ 0.05 \\ -0.89 \end{pmatrix} - 364.8 \tag{5.12}$$

5.3.3 Generating efficient Scan patterns

As expected, the number of scanline has the largest impact upon the overall coefficient of correlation. However there is still an $R^2 = 0.480$, if the number of scanlines used is not considered.

Therefore, it is possible to optimise the estimated classification performance of the chosen scanlines without necessarily increasing the number of scanlines.

Depending upon the scanning hardware and the application, different efficient scan pattern generation algorithms can be devised using Eq. 5.12 and a cost-benefit function.

Parameters that – among others – have to be taken into account for the cost function are the rotational speed, at which the scanline's inclination angle can be changed, possible inertia properties that complicate of disallow a change of the rotation's direction, and the time the measurement of single scanline consumes.

Also, the time available for the acquisition of the range-information can be limited to a fixed value, or might be determined dynamically, e.g. if an object moves too fast or is only visible for a certain time. On the other hand, a certain quality of the classification result might be required, therefore the scan pattern's estimated classification rate would have to exceed a predefined value.

Example Cost-Benefit Function

For the cost-function, it is necessary to analyse the system used for the acquisition of the range-information. The scanning system in our example shall be the scanline deflector shown in Fig. 5.21.

The scan frequency of the used 2-D laser scanner is 75Hz, or 13.3ms per scanline. The used stepper motors have a maximum rotational speed of $360°/s$, at which no inertia problem occurs due to the motors' high torque. The inclination λ_j of the outgoing scanline j is equal to the angular difference between the two mirrors, therefore the inclination angle's maximum rotational speed is $720°/s$, or 1.39ms per degree. The initial inclination angle λ_0 may be any valid angle value.

The cost-function $c(n_{SL}, \lambda_{0..n_{SL}})$ – using the number of scanlines n_{SL}, λ_0, and the used inclination angles $\lambda_{1..n_{SL}}$ as variables – can therefore be written as

$$c(n_{SL}, \lambda_{0..n_{SL}}) = n_{SL} \cdot 13.3ms + \sum_{j=1}^{n_{SL}} (\lambda_j - \lambda_{j-1}) \cdot 1.29ms \tag{5.13}$$

The regression coefficients Θ from Eq. 5.12 can be used for the benefit-function $b(\lambda_{0..n_{SL}})$, calculating the predictor variables x_{ip} from the used inclination angles $\lambda_{1..n_{SL}}$.

$$b(\lambda_{0..n_{SL}}) = \begin{pmatrix} x_1(\lambda_{0,1..j}) \\ \vdots \\ x_p(\lambda_{0,1..j}) \end{pmatrix} \cdot \begin{pmatrix} \Theta_1 \\ \vdots \\ \Theta_p \end{pmatrix} + \Theta_0 \tag{5.14}$$

The most efficient scan pattern λ_{opt} in terms of a low cost-benefit-ratio can now be determined by solving a minimisation problem

$$\lambda_{opt} = \arg min \left(\frac{c(n_{SL}, \lambda_{0..n_{SL}})}{b(\lambda_{0..n_{SL}})} \right) \tag{5.15}$$

The accuracy of the benefit-function is evaluated using a set of 300 scan patterns with five scanlines. From the 10 scan patterns predicted to be performing best by the benefit-function, 6 are also found among the 10 best scan patterns determined by performing a classification test (cf. Fig. 5.26). Moreover, the minimum classification rate of the 10 scan patterns selected by prediction is 37.5% which is still far better than the median classification rate of 25% when using 5 scanlines.

Starting from $\lambda_0 = 0°$, the most efficient scan pattern would then be #202, as it would

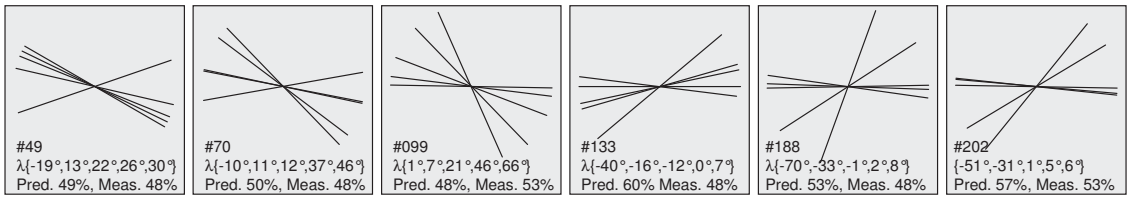


Figure 5.26: Six efficient scan patterns as selected by the benefit-function. For each scan pattern, the predicted (Pred.) and measured (Meas.) classification rate is given.

take $c = 5 \cdot 13.3ms + (6 + 57)° \cdot 1.29\frac{ms}{°} = 148ms$ to scan, resulting in a minimum cost-benefit quotient of $cb_{min}^{-1} = 2.79\frac{ms}{\%}$, as opposed to $cb^{-1} = [2.93..3.69]\frac{ms}{\%}$ for the other scan patterns.

5.4 Saliency Detection

In our system an unsupervised saliency detector is used to determine salient regions to be observed by the active vision system. Saliency information is particularly important if no traffic participants are detected, as a high saliency is indicative of traffic participants in general. At the same time, saliency detection is computationally inexpensive and thus ensures a high reactivity of the system. As input data a combination of low resolution video and 3-D motion vector information is used.

5.4.1 Implemented Saliency Detectors

Three bottom-up saliency detectors from the algorithms discussed in section 2.4.2 in the literature review are implemented: Itti *et al.* [99], Frintrop *et al.* [102], and Walker *et al.* [118]. As operators both a simple set of three Haar-like features of size 2×2 px

$$\begin{bmatrix} +1 & -1 \\ +1 & -1 \end{bmatrix}, \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \tag{5.16}$$

and Derivative of Gaussian (DoG) convolution kernels also used by Collomosse and Hall [119] shown in Fig. 5.27 are implemented and tested.

Our tests show that the differences between saliency maps generated using Haar-like features and DoG features are marginal for small kernel sizes (e.g. 3×3 px to 5×5 px) if the input is relatively noise-free.

Two ways to carry out multi-scale processing are tested. The first method is to increase

Figure 5.27: Five first and second order directional derivatives of Gaussian kernels used for feature space convolution. Source: Collomosse and Hall [119].

the sigma and matrix size of the convolution kernels to detect features of a larger scale. The second method keeps the convolution kernels constant and resizes the source image on which the convolution operation is performed. For a source image of 100×100 px $= 10^4$ px, the original convolution kernel is a 5×5 matrix. Resizing steps are $size'_k = 2 \cdot size_k - 1$ for the convolution kernel and $size'_i = 0.5 \cdot size_i$ for the source image. In the case of resizing the kernel this amounts to

$$100^2 \cdot (5^2 + 9^2 + 19^2) = 4.7 \cdot 10^6$$

operations. Resizing the image amounts to

$$(100^2 + 50^2 + 25^2) \cdot 5^2 = 3.3 \cdot 10^5$$

operations. Therefore image resizing is more than 14 times faster than kernel resizing at the chosen sizes for image and kernel. The process of resizing the image before and after the feature detection adds some computational costs to the image resizing approach, yet this is negligible in comparison to resizing the kernel.

Our tests with saliency detectors on a number of images show that the gain in detection performance by using a kernel resizing approach is minimal for both methods. The use of image resizing is proposed, as the loss in quality is minimal and the decrease of computational costs is considerable. A comparison of the resulting saliency images for both methods is shown in Fig. 5.28.

Considering that saliency is a particular reactive representation of the current environment it appears desirable to keep the delay of saliency information in the system as small as possible. The implemented saliency detectors can be adapted as to make use of the massive parallelism of field-programmable gate arrays frequently used in automotive control units.

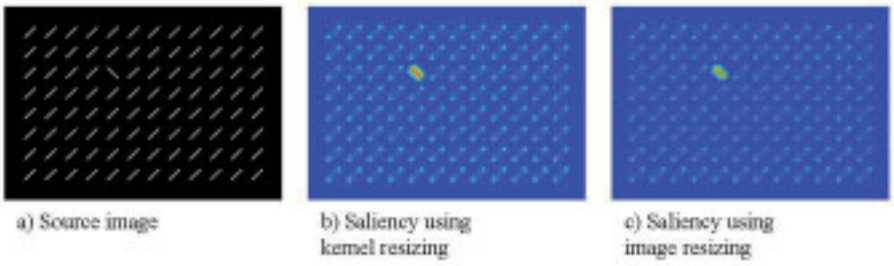


Figure 5.28: Resulting saliency images for a synthetic source image a) when using either the kernel resize method b) or the image resize method c). Image b) features a smoother transition of saliency, yet the overall quality is comparable with image c). Saliency maps are drawn in HSV colour space.

5.4.2 Evaluation of Saliency Detection

In this section, saliency algorithms are evaluated on video images using Haar-like features as operators on a Gaussian pyramid for three different spatial scales. As a measure for the quality of the determined saliency maps, the correlation coefficient between the normalised saliency map and a normalised ground truth map is used to determine the similarity.

Normalisation of all maps M is performed using

$$M_{norm} = M \left(\frac{c}{\sum_{i=1..n} M(i)} \right) - c \tag{5.17}$$

where c is a constant number. Map normalisation in Eq. 5.17 both defines the total cue strength c a saliency map can exercise in a visual attention system and ensures the expected value of M to be $\mu_M = 0$, which is necessary for calculation of the correlation coefficient.

The sample correlation coefficient of a ground truth map M_{GT} and a saliency map M_S is determined using Eq. 5.18

$$cor(M_{GT}, M_S) = \frac{1}{n-1} \left(\sum_{i=1..n} \frac{M_{GT}(i) \cdot M_S(i)}{\sqrt{M_{GT}(i)^2 \cdot M_s(i)^2}} \right) \tag{5.18}$$

Our evaluation is performed on three datasets with existing ground truth, two datasets containing faces and a third dataset containing a road traffic sequence acquired on a motorway.

Face datasets

The use of face datasets is not straightforward in the context of active vision for road traffic scenes. However, both labelled face datasets such as the Caltech Faces 1999 Dataset and OpenCV face detector cascades[4] are publicly available. The use of face datasets besides road traffic scenes therefore attempts to generalise our evaluation of saliency detection.

Caltech Faces 1999 Dataset In order to determine the saliency maps' correlation coefficient with faces in images, the *Caltech Faces 1999* dataset[5] containing 450 frontal face images with different lighting, expressions, and backgrounds is used. The images in the dataset have a resolution of 696 × 592 px in RGB colour space. The images are resized to 320 × 272 px and the colour space reduced to a single grayscale channel for our evaluation. The bounding rectangles provided in the ground truth file are converted to ellipses of the same height and width to better fit the faces' shapes. Sample images from the Caltech Faces 1999 dataset can be seen in Fig. 5.29 together with the respective saliency maps for two example frames.

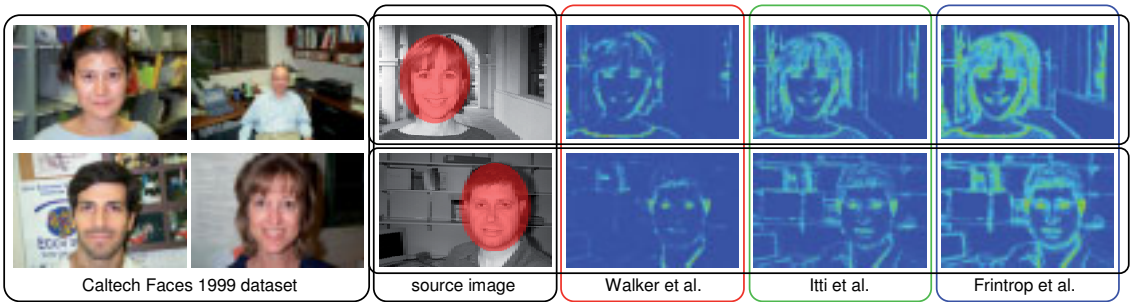


Figure 5.29: Caltech Faces 1999 dataset used for evaluation of the saliency algorithms. Ground truth information is shown as a red ellipse in the source image. The example saliency maps are drawn in HSV colour space.

It can be seen from the example saliency maps in Fig. 5.29 that the outputs of the different saliency algorithms have characteristic properties. Saliency maps generated with Walker *et al.* saliency show a good background suppression while only strong facial contours are regarded as salient. Frintrop *et al.* saliency shows a strong response in the face, yet the background is largely regarded salient as well. Itti *et al.* saliency maps appear to be balanced between the two other algorithms with regard to foreground/background distinction. Table 5.6 specifies the mean and median correlation coefficient of the saliency

[4]The OpenCV image processing library provides a set of robust face detector cascades.
[5]The Caltech Faces 1999 dataset is available: http://www.vision.caltech.edu/html-files/archive.html

maps and the ground truth maps for the Caltech Faces 1999 database.

	Correlation coefficient	
Method	Mean	Median
Walker *et al.*	0.064	0.053
Itti *et al.*	0.141	0.137
Frintrop *et al.*	0.161	0.158

Table 5.6: Correlation coefficient of saliency maps and manually labelled ground truth maps for the Caltech Faces 1999 dataset. The standard error of the mean is $SE = 0.003$ for all methods.

It can be seen from Table 5.6 that the saliency maps show only a small correlation with the ground truth maps, with the saliency map generated using Frintrop *et al.* presenting the most similar to the ground truth map, followed by Itti *et al.*and Walker *et al.* showing the smallest correlation coefficient.

Groups Dataset As a second dataset, a set of 100 images containing groups of people is used. The number of visible faces range from 3 to 14. The total number of faces visible in the image database is 805 faces which are manually labelled with bounding rectangles.

Our *Groups*[6] dataset is well suited to evaluate the performance of visual attention methods for face recognition, as group members are usually facing the camera and occlusion is rare. The area the faces cover is small as compared to the total image area, fitting the saliency detectors' scales. This is also a key difference to *Labelled Faces in the Wild* [178], a database of face photographs collected from the web for studying the problem of unconstrained face recognition, which is problematic for visual attention methods, as each face in the Faces in the Wild dataset covers a large portion of the image.

Analogous to the Caltech 1999 Faces dataset, the images are resized to 320 px in width, reduced to a single grayscale channel and the bounding rectangles provided in the ground truth file are converted to ellipses of the same height and width. Sample images from the Groups dataset can be seen in Fig. 5.30 together with the respective saliency maps for two example frames.

Similar to the results for the Caltech Faces 1999 dataset above, Walker *et al.* exhibits the best background suppression, Frintrop *et al.* responds well to the faces, and Itti *et al.* ranges in between the former two. Table 5.7 specifies the mean and median correlation coefficient of the saliency maps and the ground truth maps for the Groups database.

[6]The Groups dataset is available online: http://www.matzka.net/vision/html/datasets.html

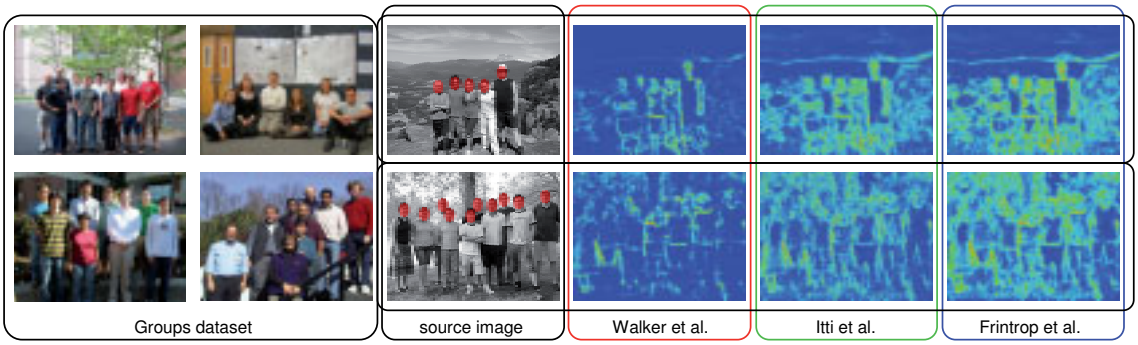


Figure 5.30: Groups dataset used for evaluation of the saliency algorithms. Ground truth information is shown as red ellipses in the source image. The example saliency maps are drawn in HSV colour space.

	Correlation coefficient	
Method	Mean	Median
Walker *et al.*	0.193	0.189
Itti *et al.*	0.203	0.211
Frintrop *et al.*	0.252	0.260

Table 5.7: Correlation coefficient of saliency maps and manually labelled ground truth maps for the Groups dataset. The standard error of the mean is $SE = 0.009$ for all methods.

The results in Table 5.7 show the same ranking for our Group dataset as for the Caltech Faces 1999 dataset. Frintrop *et al.* calculates the saliency map most similar to the ground truth map, followed again by Itti *et al.* and Walker *et al.* saliency maps without a statistically significant difference.

Road Traffic Sequence

In order to evaluate the presented saliency algorithms in a road traffic environment, a subset of our motorway sequence (MWY, cf. appendix A) is used. For 100 video frames a total of 451 traffic participants are labelled by hand to be used as ground truth. Again evaluation is performed at a lower resolution of 320 × 240 px. Sample frames from the road traffic sequence can be seen in Fig. 5.31 together with the respective saliency maps for two example frames.

Besides an evaluation of the saliency algorithms' correlation coefficient with the ground truth map, the saliency to detection cross-correlation with a trained detector is examined.

Correlation Coefficient with Ground Truth The saliency maps in Fig. 5.31 suggest that the differences between the evaluated approaches are smaller than for the face

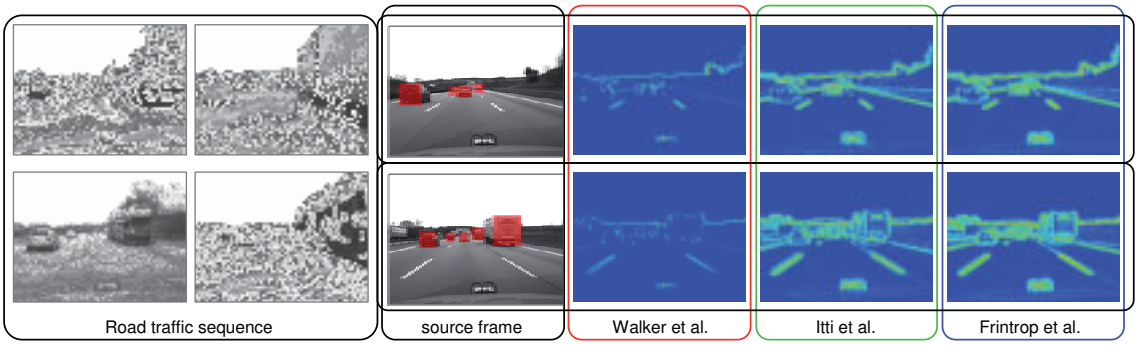


Figure 5.31: Motorway sequence (MWY) used for evaluation of the saliency algorithms. Ground truth information is shown as a red rectangles in the source image. The example saliency maps are drawn in HSV colour space.

recognition datasets. This is confirmed by the mean and median correlation coefficient of the saliency maps and the ground truth maps for the road traffic sequence in Table 5.8.

	Correlation coefficient	
Method	Mean	Median
Walker *et al.*	0.216	0.197
Itti *et al.*	0.228	0.198
Frintrop *et al.*	0.250	0.212

Table 5.8: Cross-correlation of saliency maps and manually labelled ground truth map for the motorway sequence. The standard error of the mean is $SE = 0.010$ for all methods.

Table 5.8 shows that using Frintrop *et al.* provides the highest correlation coefficient, followed by Itti *et al.* which is closely followed by the map generated using Walker *et al.*.

Cross Correlation with Detector Information Combining multiple sources of information is a classical problem in the field of data fusion. Intuitively, including additional sensor data is always beneficial or at least not detrimental to a decision process such as visual attention system, as long as the new source provides consistent information. However, this is only true for non-correlated sources (cf. Blackman and Popoli [179]). In our case both bottom-up and top-down information is determined using the same video data of the road traffic scene, so a cross-correlation of saliency and detection is likely.

In Tab. 5.8 the cross correlation of saliency as bottom-up information with a ground truth map is evaluated. In order to assess the additional information provided to the information fusion process by saliency, the correlation with other information sources must be considered. As in the systems proposed in Navalpakkam and Itti [137] and Frintrop [103], this is a set of trained classifiers.

A trained vehicle detector is applied on our motorway sequence to generate a normalised detection map similar to the ground truth maps. In Table 5.9 the mean and median correlation coefficients of the saliency maps and the detection maps are given for the motorway sequence.

	Correlation coefficient	
Method	Mean	Median
Walker *et al.*	0.117	0.114
Itti *et al.*	0.170	0.177
Frintrop *et al.*	0.181	0.181

Table 5.9: Correlation coefficient of saliency map map and label map generated by a car detector. The standard error of the mean is $SE = 0.009$ for all methods.

The correlation coefficients with the trained classifiers results in Table 5.9 are significant, yet not as high as the correlation, and therefore the similarity, with the ground truth in Table 5.8. Two things can be inferred from from this: First, the similarity to the ground truth map is higher than the similarity to the detector information. This is indicative of additional information that can be gained by using saliency as a bottom-up information. Second, the amount of correlation between the given sources is known, which can be considered in the information fusion process (cf. Blackman and Popoli [179] and references therein).

5.5 Time-to-collision

Time-to-collision (TTC) is described as an effective measure to assess the severity of road-traffic conflicts by van der Horst and Hogema [180]. In the literature, TTC has been defined as

> *"...the time required for two vehicles to collide if they continue at their present velocity and on the same path."* (Hayward [181])

Since then, this definition has been broadened to allow for changes in velocity and path using tracking systems (cf. Blackman and Popoli [179]). Methods to determine TTC from a video image are described in the literature, e.g. Galbraith *et al.* [182]. TTC is estimated by combining the range measurements acquired by a laser scanner and the range profile motion presented in section 4.5.1. In order to account for the dymanics of both ego-vehicle and other traffic participants, a maximum constant acceleration of

$$a_{max} = -2.5\frac{m}{s^2} - 1.8\frac{m}{s^2} = -4.3\frac{m}{s^2}$$

relative to the ego-vehicle is assumed considering the positive ($1.8\frac{m}{s^2}$) and negative ($-2.5\frac{m}{s^2}$) maxima of the acceleration envelope shown in Fig. 5.32. This reflects the situation of the ego-vehicle accelerating with the 85 percentile acceleration and the observed traffic participant slowing down with the 85 percentile deceleration given in Fig. 5.32. Therefore the chosen value for a_{max} is sufficient to the vast majority of all road traffic situations. In the few cases where a_{max} underestimates the actual object dynamics, the changing relative velocity $\tilde{v}$ is able to compensate this effect well due to laser scanner's high sample rate of 75 Hz.

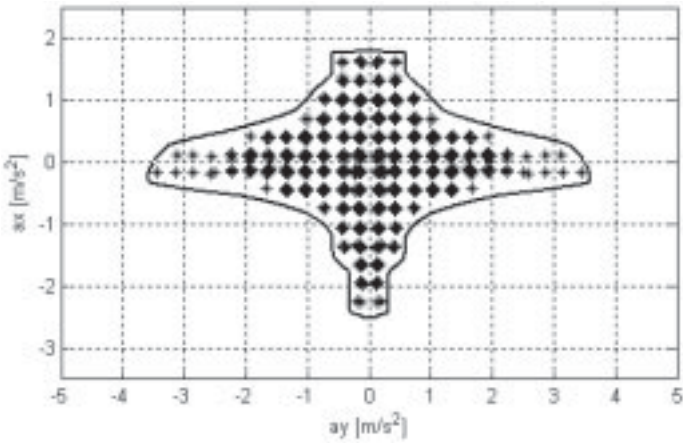


Figure 5.32: Acceleration envelope (85 percentile) of a 2.5 hour test drive on urban roads, country roads, and motorways in x-direction and y-direction. Source: Wegscheider and Prokop [183].

A simple motion model assuming constant velocity and acceleration is used to estimate t'_{TTC} (cf. Eq. 5.19 and 5.20).

$$r = \tilde{v} \cdot t'_{TTC} - \frac{a}{2} \cdot (t'_{TTC})^2 \tag{5.19}$$

$$t'_{TTC_{1,2}} = \frac{-\tilde{v} \pm \sqrt{\tilde{v}^2 - 2 \cdot a_{max} \cdot r}}{a_{max}} \tag{5.20}$$

Due to the chosen negative acceleration a_{max} and positive range values r, the quadratic formula in Eq. 5.20 always returns two solutions $t'_{TTC_{1,2}}$ of which at least one is positive. The relevant t_{TTC} is then defined as the minimum positive t'_{TTC} value and is determined

using Eq. 5.21.

$$t_{TTC} = \begin{cases} t'_{TTC_1} & \text{if } t'_{TTC_2} < 0s \\ t'_{TTC_1} & \text{if } 0s \leq t'_{TTC_1} \leq t'_{TTC_2} \\ t'_{TTC_2} & \text{otherwise} \end{cases} \tag{5.21}$$

In Tab. 5.10 a set of calculated values for t_{TTC} for r =[2.5 m,30 m], $\tilde{v} = [-5\frac{m}{s}, 5\frac{m}{s}]$, and $a_{max} = -4.3\frac{m}{s^2}$ is given. The TTCs for these typical range and velocity values are between 0.42 s and 5.08 s.

	r[m]					
$\tilde{v}[\frac{m}{s}]$	2.5	5	10	15	20	30
+5	2.75	3.08	3.61	4.05	4.43	5.08
+3	1.98	2.37	2.96	3.43	3.83	4.50
0	1.08	1.52	2.16	2.64	3.05	3.74
-3	0.59	0.98	1.57	2.03	2.43	3.10
-5	0.42	0.75	1.29	1.72	2.10	2.75

Table 5.10: Time-to-collision values in seconds for given relative velocities $\tilde{v}$ and distances r assuming a constant relative acceleration of $4.3\frac{m}{s^2}$ towards the ego-vehicle.

A motorway sequence with overlayed time-to-collision information estimated using laser range data on video data is given[7]. An example frame of the video can be seen in Fig. 5.33a. A plan view representation of the same scene is given in Fig. 5.33b.

a) 

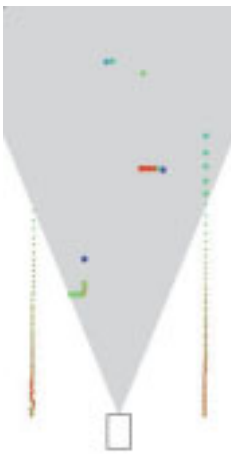
b) 

Figure 5.33: Overlay of time-to-collision information estimated using laser range data on video data shown in a). Figure b) shows the same scene as seem in plan view. Colour of range readings indicates the estimated time-to-collision.

[7]The MWY sequence with overlayed time-to-collision information is available online: http://www.matzka.net/vision/html/motorway.html

5.6 Discussion of Semantic Level Modules

Traffic participant detection and classification are the central modules in the semantic level as the resulting information is used both by driver assistance systems and the contextual resource allocation. Situated in the third level of the proposed system, traffic participant information is provided to driver assistance systems with minimal latency. The reinforcing relationship between information acquisition and resource allocation is pointed out in section 7.1.1.

The trained cascades provide a robust traffic participant detection and classification for a wide range of traffic participant types. However, due to the prototypical character of our proposed system, a number of limitations exist:

- Classifier cascades do not cover all groups of traffic participants.
- Cascades are trained on cars and lorries using only rear views.
- All training samples are acquired from daylight scenes.

While bicycles and light motorcycles are detected by our human detector cascade, no classifier cascades for either bicycles or motorcycles are trained. This is due to a lack of positive samples in our acquired test sequences.

For our human detector cascade and pedestrian cascades positive samples cover a large range of viewpoints, e.g. frontal, side, and back views. This is possible due to the partial viewpoint independence of human shapes especially of the shoulder line and the lower torso. However both cars and lorries are not viewpoint invariant. The training of car cascades and lorry cascades is limited to using only rear views, effectively precluding the correct detection and classification of cars and lorries from other viewpoints. This limitation is in compliance with our used sensors' detection ranges discussed in section 3.5.2. There it is argued that the time-to-collision of vehicle traffic participants driving in the opposite direction is too short for our resource allocation system.

All of our training samples are acquired from daylight scenes resulting in daylight-specific distinctive features encoded in the cascades. A prominent example for this is the shadow under a vehicle in daylight, which is commonly used as a feature for vehicle detection (e.g. Mori and Charkari [184], Sun *et al.* [185]). This shadow is not always visible in night-time scenes (cf. Fig. 5.34). While the detection of a vehicle's shadow

is not used as a detection concept in our proposed system, the trained cascades contain corresponding Haar-like features as a result of using daylight training samples.

a)  b) 

Figure 5.34: Shadow feature used as a cue for vehicle detection. While the shadow under a car is a good feature in daylight conditions a), it is not always visible in night-time scenes such as b).

Chapter 6

Reasoning Level

In the proposed system's reasoning level, semantic information about the environment is gathered, processed, and fused. Both current context and candidate regions to be observed are provided to the contextual resource allocation level. For this, the determination and fusion of traffic participant probabilities is described in section 6.1. A candidate region determination module described in section 6.2 identifies candidate regions to be observed by the active vision system. In section 6.3 a discussion of the reasoning level modules is given.

6.1 Traffic Participant Probability Determination

In this section the determination of traffic participant probabilities is described. For this, statistical information in section 6.1.1 and dynamic information obtained using both prior knowledge and detected traffic participants in section 6.1.2 are fused in section 6.1.3 using a covariance union method. The proposed concept is illustrated in Fig. 6.1.

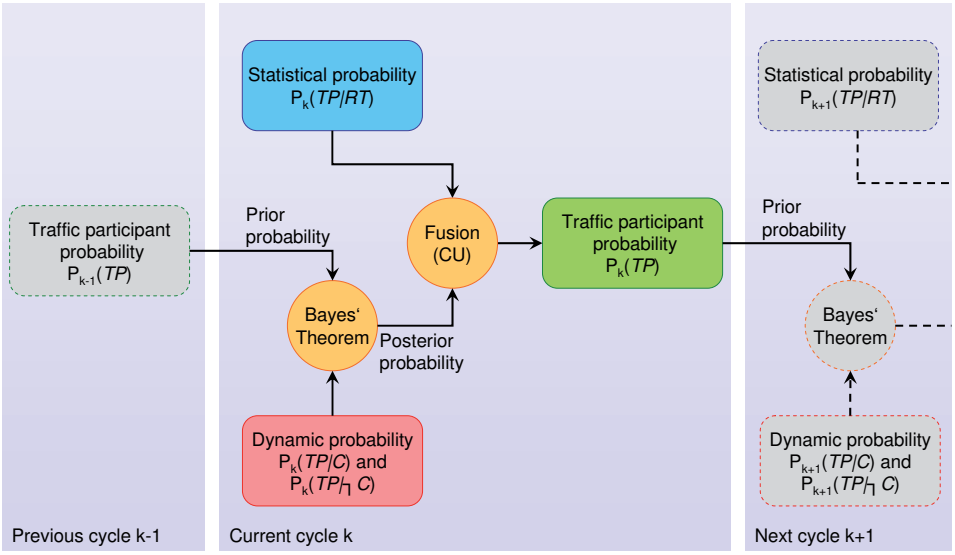


Figure 6.1: Probability concept for determining traffic participant probabilities.

6.1.1 Statistical Information

Statistical information about the ego-vehicle's environment can be gained by considering the current road type and the specific situation associated with it. Statistical data provided in the annual survey of road accidents of the Federal Statistical Office Germany [186] provides a good basis to determine the relative frequency and severity of accidents for different traffic participant types TP_n and for different road types RT_m. For 2006, the fraction of traffic participants injured in accidents with cars is given in Tab. 6.1.

Road type	Pedestrian	Bicycle	Motorcycle	Car	Lorry
Urban traffic	0.088	0.175	0.092	0.606	0.039
Country road	0.018	0.047	0.095	0.722	0.117
Motorway	0.005	0.001	0.025	0.732	0.236

Table 6.1: Fraction of traffic participants injured in accidents with cars in 2006 differentiated by type of traffic participation and road types. Source: Federal Statistical Office Germany [186], Tab. UJ 5 (1-4).

The fraction of injured traffic participants in Tab. 6.1 is used as the conditional probability $P(\mathit{TP}_n|\mathit{RT}_m)$ of a traffic participant type TP_n to exist on a certain road type RT_m.

6.1.2 Dynamic Information

Statistical information based on road type alone is used as long as no current knowledge about traffic participants in the environment exists. Therefore it is valid to assume that the probability of a pedestrian to appear on a motorway is almost zero. However, as soon as a detected object is classified to be a pedestrian, this assumption does not hold.

For each detector cascade and classifier cascade $\mathcal{C}$ a probability of $\mathcal{TP}_n$ dependent upon a positive classifier result C or negative classifier result $\neg C$ exists. As both positive and negative results can either be true or false, there exist four distinguishable cases:

$$P(\mathcal{TP}_n|C),\ P(\mathcal{TP}_n|\neg C),\ P(\neg\mathcal{TP}_n|C),\ \textit{and}\ P(\neg\mathcal{TP}_n|\neg C)$$

In order to determine the probability of $P(\mathcal{TP}_n)$ to exist dependent upon positive C, or negative $\neg C$ classifier results, Bayes' theorem is used. The individual probabilities are calculated using Eq. 6.1–6.4

$$P_k(\mathcal{TP}_n|C) = \frac{P(C \mid \mathcal{TP}_n)P_{k-1}(\mathcal{TP}_n)}{P(C)} \tag{6.1}$$

$$P_k(\mathcal{TP}_n|\neg C) = \frac{P(\neg C \mid \mathcal{TP}_n)P_{k-1}(\mathcal{TP}_n)}{P(\neg C)} \tag{6.2}$$

$$P_k(\neg\mathcal{TP}_n|C) = \frac{P(C \mid \neg\mathcal{TP}_n)P_{k-1}(\neg\mathcal{TP}_n)}{P(C)} \tag{6.3}$$

$$P_k(\neg\mathcal{TP}_n|\neg C) = \frac{P(\neg C \mid \neg\mathcal{TP}_n)P_{k-1}(\neg\mathcal{TP}_n)}{P(\neg C)} \tag{6.4}$$

where $P(C)$ and $P(\neg C)$ are normalising constants ensuring a probability sum of 1.00 and can be determined using Eq. 6.5 and 6.6.

$$P(C) = P_k(C \mid \mathcal{TP}_n)P_{k-1}(\mathcal{TP}_n) + P_k(C \mid \neg\mathcal{TP}_n)P_{k-1}(\neg\mathcal{TP}_n) \tag{6.5}$$

$$P(\neg C) = 1 - P(C) \tag{6.6}$$

For Eq. 6.1–6.6 the true positive rates, false positive rates, true negative rates, and false negative rates of all detectors and classifiers must be known as well as the prior

probability $P_{k-1}(\mathcal{TP}_n)$. From detector and classifier training, true positive rates and false positive rates are known. The negative rates are then calculated using Eq. 6.7 and 6.8.

$$P(\neg C \mid \mathcal{TP}_n) = 1 - P(C \mid \mathcal{TP}_n) \tag{6.7}$$

$$P(\neg C \mid \neg\mathcal{TP}_n) = 1 - P(C \mid \neg\mathcal{TP}_n) \tag{6.8}$$

The true positive rate is governed by the detector and classifier cascades' properties due to the comparably small number of traffic participants in a single image. False positives rates are also considerably dependent upon the image size, as discussed in section 5.2.5. A false positive rate of the $P(C|\neg\mathcal{TP}) = 10^{-7}$ per sample then results in the probability of

$$P(C \mid \neg\mathcal{TP}) = 1 - \left(1 - 10^{-7}\right)^{10^6} = 0.095$$

for a false positive considering 10^6 examined samples inside a single video frame.

The posterior probabilities $P_k(\mathcal{TP}_n|C)$ and $P_k(\mathcal{TP}_n|\neg C)$ for varying values of $P_{k-1}(\mathcal{TP})$, $P(C|\mathcal{TP}_n)$, and $P(C|\neg\mathcal{TP}_n)$ are given in Fig. 6.2.

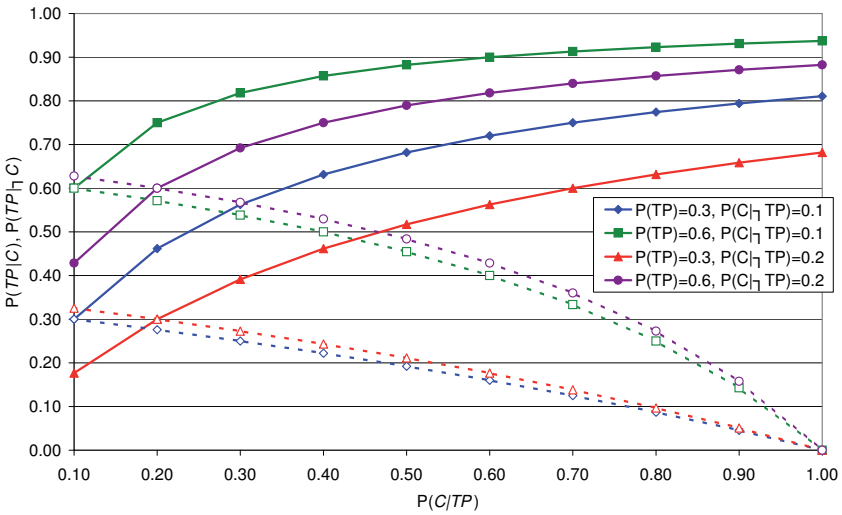


Figure 6.2: Probabilities $P_k(\mathcal{TP}_n|C)$ for a true positive (solid) and $P_k(\mathcal{TP}_n|\neg C)$ for a false negative (dotted) for varying values of $P_{k-1}(\mathcal{TP}_n) = (0.3; 0.6)$, $P(C|\mathcal{TP}_n) = [0.1, 1.0]$, and $P(C|\neg\mathcal{TP}_n) = (0.1; 0.2)$.

From Fig. 6.2 it can be seen that the probability $P_k(\mathcal{TP}_n|C)$ increases and $P_k(\mathcal{TP}_n|\neg C)$ decreases for a stronger classifier $\mathcal{C}$ and thus an increasing true positive rate $P(C|\mathcal{TP}_n)$. For increasing false positive rates $P(C|\neg\mathcal{TP}_n)$, the reliability of a positive classification suffers, decreasing $P_k(\mathcal{TP}_n|C)$. Considering the graphs for true positive rates (solid) and

false negative rates (dotted), the difference $\Delta P(\mathcal{TP}_n) = P_k(\mathcal{TP}_n|C) - P_k(\mathcal{TP}_n|\neg C)$ between graphs can be considered to represent the actual information content provided by the detector or classifier. The determination of traffic participant probability is performed for all proposed candidate regions $\mathcal{R}_n$ and for the complete video frame $\mathcal{R}_\forall$.

It must be noted that the sum of all posterior traffic participant type probabilities $\sum P_k(\mathcal{TP}_n)$ is not necessarily 1.00. This is due to the independence of traffic participant probabilities of different types. If no traffic participant is detected, the sum of all posterior traffic participant type probabilities is likely to be less than 1.00. If multiple traffic participants of the same types are detected, the sum of posterior traffic participant probabilities is likely to be more than 1.00.

Decomposition of Traffic Participant Probabilities

In our proposed system it is necessary to decompose the traffic participant probabilities $P(\mathcal{TP}_H)$ and $P(\mathcal{TP}_V)$ determined using the detector cascades $\mathcal{C}_H$ and $\mathcal{C}_V$ into probabilities for individual traffic participant types $P(\mathcal{TP}_{1,..,5})$. This decomposition is performed using the statistical traffic participant probability on road types $P(\mathcal{TP}_n|\mathcal{RT}_m)$.

$$P(\mathcal{TP}_n|\mathcal{RT}_m, P(\mathcal{TP}_H|C)) = P(\mathcal{TP}_H|C) \cdot \frac{P(\mathcal{TP}_n|\mathcal{RT}_m)}{P(\mathcal{TP}_H|\mathcal{RT}_m)},\ n \in 1,2,3 \tag{6.9}$$

$$P(\mathcal{TP}_n|\mathcal{RT}_m, P(\mathcal{TP}_V|C)) = P(\mathcal{TP}_V|C) \cdot \frac{P(\mathcal{TP}_n|\mathcal{RT}_m)}{P(\mathcal{TP}_V|\mathcal{RT}_m)},\ n \in 4,5 \tag{6.10}$$

where

$$P(\mathcal{TP}_H|\mathcal{RT}_m) = P(\mathcal{TP}_1|\mathcal{RT}_m) + P(\mathcal{TP}_2|\mathcal{RT}_m) + P(\mathcal{TP}_3|\mathcal{RT}_m) \tag{6.11}$$

$$P(\mathcal{TP}_V|\mathcal{RT}_m) = P(\mathcal{TP}_4|\mathcal{RT}_m) + P(\mathcal{TP}_5|\mathcal{RT}_m) \tag{6.12}$$

In our proposed system, the statistical traffic participant probability $P(\mathcal{TP}_n|\mathcal{RT}_m)$ is used to decompose traffic participant probabilities instead of using the prior traffic participant probability $P_{k-1}(\mathcal{TP}_n)$. This appears problematic, as statistical knowledge is also used in the fusion of statistical and dynamic knowledge described in section 6.1.3 below, resulting in a covariance between statistical information and dynamic information. However, the proposed system uses statistical information considering three aspects.

First, no dynamic information for both bicycles $\mathcal{TP}_2$ and motorcycles $\mathcal{TP}_3$ is available due to the lack of classifiers for either traffic participant type. Second, classifier information is dependent upon the allocation of classifier processes in the previous cycle. This induces a reinforcing relationship between the allocation of classifiers and the prior traffic participant probability influenced by classifier results. Third, the covariance union method discussed in section 6.1.3 below is robust towards covariant input data.

6.1.3 Fusion of Statistical and Dynamic Information

Statistical and dynamic information is fused to determine robust traffic participant probabilities $P_k(\mathcal{TP}_n)$ for all candidate regions and the complete video frame. However, a fusion of statistical and dynamic information is problematic if the data is inconsistent.

An example for this is the repeated detection of a human traffic participant on a motorway, resulting in a high dynamic probability for pedestrians, bicycles, and motorcycles (e.g. $P_k(\mathcal{TP}_1|C, P_{k-1}(\mathcal{TP}_1)) = 0.800$). In contrast to this, the statistical probability for pedestrians is small with $P(\mathcal{TP}_1|\mathcal{RT}_5) = 0.015$. Assuming the variances of these probability values to be $varP(\mathcal{TP}_1|\mathcal{RT}_5) = 0.3^2$, and $varP_k(\mathcal{TP}_1|C, P_{k-1}(\mathcal{TP}_1)) = 0.1^2$, the probability distributions are drawn in Fig. 6.3.

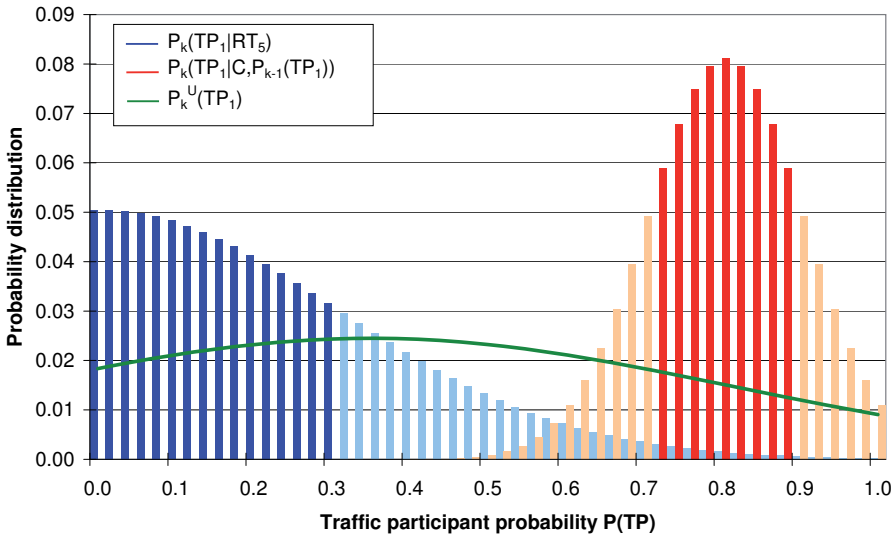


Figure 6.3: Inconsistency between statistical probability (blue bars) and dynamic probability (red bars). Probability values within $\mu \pm 1\sigma$ are drawn with a higher saturation. The fused traffic participant probability $P_k^{\cup}(\mathcal{TP})$ (green line) is determined using the covariance union method.

In Fig. 6.3 the difference between $P_k(\mathcal{TP}_1|C, P_{k-1}(\mathcal{TP}))$ and $P(\mathcal{TP}_1|\mathcal{RT}_5)$ cannot be accounted for by the assumed variance of traffic participant probabilities. In Matzka and Altendorfer [12, 16] three information fusion methods are discussed. Of these, the covariance union method first proposed by Uhlmann [187] presents a method for fault-tolerant data fusion. In our proposed system, the covariance union method is used to determine both the optimum fused traffic participant probability $P_k^{\cup}(\mathcal{TP}_1)$, and a conservative estimate of the traffic participant probability's variance. An illustration of the covariance union method is given in Fig. 6.4.

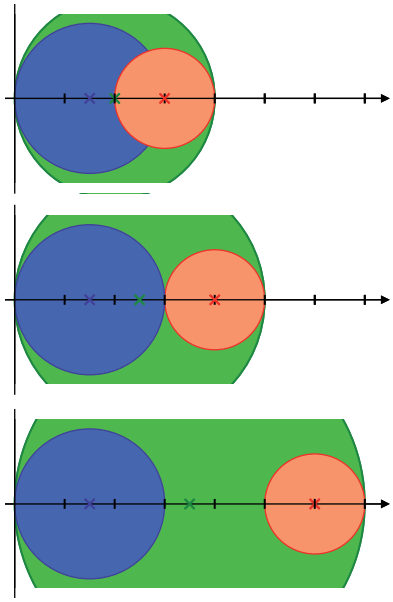


Figure 6.4: Illustration of the covariance union method for fault-tolerant data fusion. As the distance between the two estimates (red, blue) with different variances increases, the fused estimate and variance of the fusion result (green) adapts to include both estimates and their respective variances.

While the reader is referred to the original article by Uhlmann [187] for a more detailed discussion of the covariance union method, the used equations are given in the following. The traffic participant probability vector $P_k^{\cup}(\mathcal{TP})$ is found by minimising the trace of its covariance matrix

$$P_k^{\cup}(\mathcal{TP}) = \arg\min\left(\mathrm{tr}\left(varP_k^{\cup}(\mathcal{TP})\right)\right) \tag{6.13}$$

where $varP_k^{\cup}(\mathcal{TP})$ is determined as

$$varP_k^{\cup}(\mathcal{TP}) = max(U_a, U_b) \tag{6.14}$$

with

$$U_a = varP(\mathcal{TP}|\mathcal{RT}) + (P_k^{\cup}(\mathcal{TP}) - P(\mathcal{TP}|\mathcal{RT})) \cdot (P_k^{\cup}(\mathcal{TP}) - P(\mathcal{TP}|\mathcal{RT})^T \tag{6.15}$$

$$U_b = varP_k(\mathcal{TP}|C, P_{k-1}(\mathcal{TP})) \tag{6.16}$$

$$+ (P_k(\mathcal{TP}|C, P_{k-1}(\mathcal{TP})) - P_k^{\cup}(\mathcal{TP})) \cdot (P_k^{\cup}(\mathcal{TP}) - P_k(\mathcal{TP}|C, P_{k-1}(\mathcal{TP})))^T \tag{6.17}$$

For the example given in Fig. 6.3 the covariance union method results in a $P_k^{\cup}(\mathcal{TP}_1) = 0.350$ with a variance of $varP_k^{\cup}(\mathcal{TP}_1) = 0.461^2$. This result is also drawn in in Fig. 6.3 as a green graph. As an example for a consistent pair of statistical and dynamic probabilities, we assume the probabilities of a car to exist to be

$$P(\mathcal{TP}_3|\mathcal{RT}_5) = 0.6,\ varP(\mathcal{TP}|\mathcal{RT}) = 0.3^2$$

$$P_k(\mathcal{TP}_3|C, P_{k-1}(\mathcal{TP}_3)) = 0.8,\ varP_k(\mathcal{TP}_3|C, P_{k-1}(\mathcal{TP}_3)) = 0.1^2$$

resulting in a $P_k^{\cup}(\mathcal{TP}_3) = 0.6$ with a variance of $varP_k^{\cup}(\mathcal{TP}_3) = 0.3^2$. This result shows that the covariance union method is aimed to provide a fault-tolerant, conservative estimate as opposed to a covariance intersection method proposed by Julier and Uhlmann [188], where the smaller variance of $P_k(\mathcal{TP}_3|C, P_{k-1}(\mathcal{TP}_3))$ dominates the fusion result. For our proposed system, fault-tolerance is more important than small resulting variances due to the frequent occurrence of inconsistent probability pairs.

Example Fusion Process

As an example for our fusion of statistical and dynamic information, we assume an urban environment $\mathcal{RT}_3$, two detected human traffic participants $N_{\mathcal{TP}_H} = 2$, and no detected vehicles $N_{\mathcal{TP}_V} = 0$. The prior traffic participant probability $P_{k-1}(\mathcal{TP})$ determined in the previous cycle $k-1$ is assumed as

$$P_{k-1}(\mathcal{TP}) = (0.150, 0.200, 0.150, 0.500, 0.100)$$

From this prior traffic participant probability the probabilities for both $P_{k-1}(\mathcal{TP}_H) = 0.500$, and $P_{k-1}(\mathcal{TP}_V) = 0.600$ are calculated using Eq. 6.11 and 6.12. Using Bayes'

theorem given in Eq. 6.1 to 6.8 the posterior probability $P_k(\mathcal{TP}_H|2C)$ and $P_k(\mathcal{TP}_V|\neg C)$ are obtained

$$P_k(\mathcal{TP}_H|2C) = 1 - (1 - 0.842)^2 = 0.975, \quad [P(C|\mathcal{TP}_H) = 0.800, P(C|\neg\mathcal{TP}_H) = 0.150]$$

$$P_k(\mathcal{TP}_V|\neg C) = 0.200, \quad [P(C|\mathcal{TP}_V) = 0.850, P(C|\neg\mathcal{TP}_V) = 0.100]$$

assuming conservative values for $P(C|\mathcal{TP}_x)$ and $P(C|\neg\mathcal{TP}_x)$. The posterior probabilities $P_k(\mathcal{TP}_H|2C)$ and $P_k(\mathcal{TP}_V|\neg C)$ are then decomposed into individual traffic participant type probabilities using Eq.6.9 to 6.12.

$$P_k(\mathcal{TP}_1|C, P_{k-1}(\mathcal{TP}_1)) = 1 - \left(1 - \left(0.842 \cdot \frac{P(\mathcal{TP}_1|\mathcal{RT}_3)}{P(\mathcal{TP}_H|\mathcal{RT}_3)}\right)^2\right) = 0.374$$

$$P_k(\mathcal{TP}_2|C, P_{k-1}(\mathcal{TP}_2)) = 1 - \left(1 - \left(0.842 \cdot \frac{P(\mathcal{TP}_2|\mathcal{RT}_3)}{P(\mathcal{TP}_H|\mathcal{RT}_3)}\right)^2\right) = 0.658$$

$$P_k(\mathcal{TP}_3|C, P_{k-1}(\mathcal{TP}_3)) = 1 - \left(1 - \left(0.842 \cdot \frac{P(\mathcal{TP}_3|\mathcal{RT}_3)}{P(\mathcal{TP}_H|\mathcal{RT}_3)}\right)^2\right) = 0.389$$

$$P_k(\mathcal{TP}_4|C, P_{k-1}(\mathcal{TP}_4)) = 0.200 \cdot \frac{P(\mathcal{TP}_4|\mathcal{RT}_3)}{P(\mathcal{TP}_V|\mathcal{RT}_3)} = 0.188$$

$$P_k(\mathcal{TP}_5|C, P_{k-1}(\mathcal{TP}_5)) = 0.200 \cdot \frac{P(\mathcal{TP}_5|\mathcal{RT}_3)}{P(\mathcal{TP}_V|\mathcal{RT}_3)} = 0.012$$

The fusion of statistical knowledge and dynamic knowledge is then performed using the covariance union method (cf. Eq. 6.13 to 6.17) with

$$P_k(\mathcal{TP}|C, P_{k-1}(\mathcal{TP})) = (0.374, 0.658, 0.389, 0.188, 0.012),\ var(P_k(\mathcal{TP}|C, P_{k-1}(\mathcal{TP}))) = 0.1^2 \cdot I_5$$

$$P(\mathcal{TP}|\mathcal{RT}_3) = (0.088, 0.175, 0.092, 0.606, 0.039),\ var(P(\mathcal{TP}|\mathcal{RT}_3)) = 0.3^2 \cdot I_5$$

which results in a fused traffic participant probability $P_k^{\cup}(\mathcal{TP})$ of

$$P_k^{\cup}(\mathcal{TP}) = (0.183, 0.320, 0.185, 0.302, 0.012)$$

6.2 Candidate Region Determination

Performing resource allocation requires that proposals for regions to be observed are collected, merged and extended, and that their respective utility is determined. Fig. 6.5 depicts the processing steps towards the actual resource allocation illustrated in Fig. 7.1.

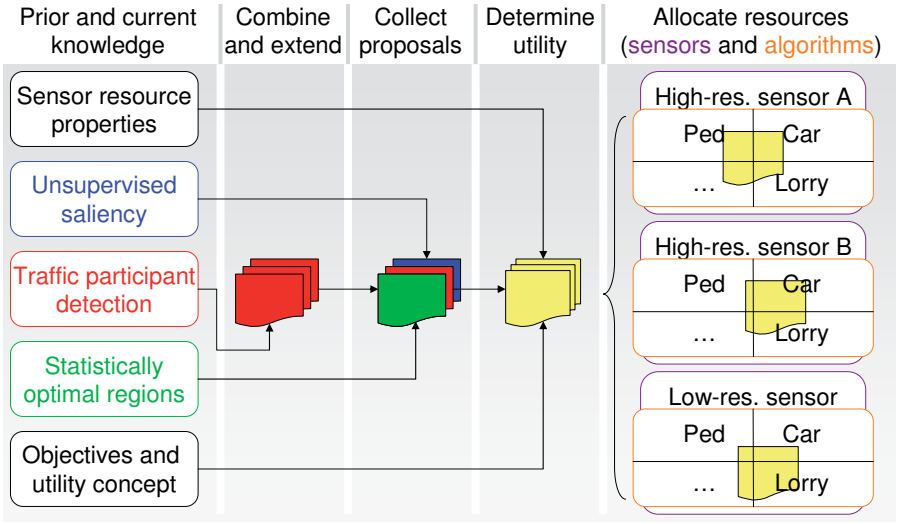


Figure 6.5: Resource allocation scheme. Proposals for regions to be observed are collected using salient regions (section 6.2.1), extended and merged regions with detected traffic participants (section 6.2.2), and static regions (section 6.2.3). The overall combined utility is determined for each candidate region (section 7.2). Selected regions are allocated a sensor-algorithm combination that is estimated to yield the best result (section 7.3 and 7.4).

Candidate regions $\mathcal{R}$ contain information about the upper-left corner coordinates (i_{UL}, j_{UL}) and lower-right corner coordinates (i_{LR}, j_{LR}) as well as the probability of traffic participant types $P(\mathcal{TP}_n)$, the region's saliency S, and current uncertainty $\mathcal{UC}$.

$$\mathcal{R} = \begin{pmatrix} i_{UL} \\ j_{UL} \\ i_{LR} \\ j_{LR} \\ P(\mathcal{TP}_n) \\ S \\ \mathcal{UC} \end{pmatrix}$$

6.2.1 Use of Saliency

Several candidate regions are determined from the saliency map provided by the saliency detection module. The number of candidate regions is either a predetermined number $N_{\mathcal{R}}$ or is determined using the saliency maps statistics, e.g. the number of local maxima. Our proposed system is evaluated using $N_{\mathcal{R}}$=[1,5] salient regions. These are determined by iteratively selecting the region with the highest overall saliency using an integral image representation of the downscaled saliency map. The algorithm to determine candidate regions from the saliency maps is as follows:

1. Downscale saliency map to 64×48 px to increase search speed.
2. Calculate an integral image representation of the saliency map (cf. section 2.2.1).
3. Determine the 16×12 px region with the highest overall saliency using the integral image of the saliency map to increase search speed.
4. The region with highest overall saliency is used as a candidate region.
5. The saliency map is attenuated by a factor of 0.5 inside the candidate region.
6. Repeat steps 2.–5. until the desired number of candidate regions is found.

The above algorithm, which is also illustrated in Fig. 6.6, is computationally inexpensive while providing a better performance than two other algorithms implemented and tested on our proposed system: maxima elimination and clustering.

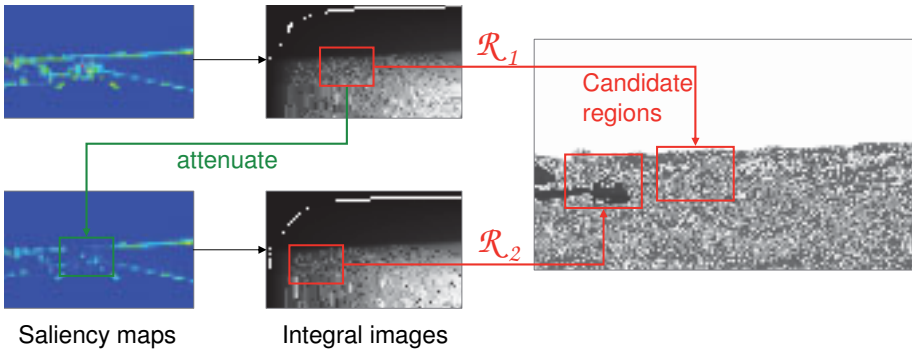


Figure 6.6: Candidate region determination using saliency information. The downscaled saliency map is converted to an integral image representation, where the region with maximum overall saliency is searched. Selected regions' saliency values are attenuated before the next region is determined.

Maxima elimination is an iterative process of finding the global maximum, selecting this point as the candidate region's centre, and attenuating the maximum and its immediate

neighbourhood as in our proposed method. A disadvantage of this method is that the borders of traffic participants are generally the most salient regions of traffic participants. This leads towards candidate regions being centred on the traffic participants' borders as opposed to the traffic participants' centres.

Clustering is evaluated as a second alternative to determine regions from the saliency map. Using Lloyd's algorithm [189], a common form of the k-means clustering algorithm [190] as implemented in the *KMlocal* library [191][1]. The desired number of candidate regions $N_{\mathcal{R}}$ also defines the number of clusters. The number of clusters has a large influence on the clustering results, causing a lack of robustness. At the same time, clustering is a computationally expensive method as compared to using an integral image approach or maxima elimination.

6.2.2 Use of Traffic Participant Detection

The traffic participant detection module provides a list of regions containing detected traffic participants. The list of detected traffic participants then undergoes a merging and extension process performed with respect to sensor resource properties, in particular the aperture angle.

Merging of Candidate Regions

Unlike the human eye, where the fovea centralis covers as little as 1° in highest visual acuity (cf. section 2.4.1), the aperture angle of high-resolution sensors in the proposed system is 10° or more. If two or more candidate regions are both small and close enough to be observed at the same time by a high-resolution sensor, the minimum bounding rectangle around these candidate regions is added to the list of candidate regions. The algorithm to perform the region merging process illustrated in Fig. 6.7 is given in Algorithm 6.1.

In order to keep the list of candidate regions as small as possible, both regions and combined regions entirely included in a larger combined region can be removed from the list without loss of coverage of all traffic participants. This removal process is performed using the region's indices, as all regions with indices that are a subset of any other region's indices can be removed. For the example regions given in Fig. 6.7, all candidate regions except $\mathcal{R}_{123}$ are removed.

[1] *KMlocal* library is an open-source k-means clustering library and available online at: http://www.cs.umd.edu/~mount/Projects/KMeans/ [191].

Figure 6.7: Example for region merging. In (a) three cars are detected (regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$). These can be merged into three bounding rectangles containing two cars (b, regions $\mathcal{R}_{12}, \mathcal{R}_{13}, \mathcal{R}_{23}$), and into one bounding rectangle containing three cars (c, region $\mathcal{R}_{123}$). The combined regions found in (b) and (c) are then added to the original list of candidate regions from (a).

Input: A set of $N_\mathcal{R}$ candidate regions $\mathcal{R}_\iota$
Upper left $(i,j)_{UL}$ and lower right $(i,j)_{LR}$ corner coordinates
Output: A candidate region list $\mathcal{R}$ of length $N_\mathcal{R}$

for $m \leftarrow 1$ **to** $((N_\mathcal{R})^2 - 1)$ **do**
  for $n \leftarrow 0$ **to** $N_\mathcal{R}$ **do**
    if $(((m/2^n) \mod 2) = 1)$ **then**
      $i_{UL} = \min(i_{UL}, i_{UL}(\mathcal{R}_n))$;
      $j_{UL} = \min(j_{UL}, j_{UL}(\mathcal{R}_n))$;
      $i_{LR} = \max(i_{LR}, i_{LR}(\mathcal{R}_n))$;
      $j_{LR} = \max(j_{LR}, j_{LR}(\mathcal{R}_n))$;
    end
  end
  if $((i_{LR} - i_{UL}) < i_{max}$ & $(j_{LR} - j_{UL}) < j_{max})$ **then**
    Append new $\mathcal{R}_\iota$ with $((i,j)_{UL}, (i,j)_{LR})$ to $\mathcal{R}$;
  end
end

Algorithm 6.1: Region merging algorithm.

Extension of Candidate Regions

After merging, candidate regions are extended to match the aperture angle of high-resolution sensors. Extension is not performed on regions that exceed the high-resolution sensors' aperture angles. Instead, these large regions are either dismissed, or considered for traffic participant classification using the low-resolution data as sensor input. The latter is often the case for lorries or near pedestrians, as these frequently exceed the maximum height that can be observed by high-resolution sensors.

The extension of candidate regions is performed horizontally and vertically towards the desired aperture angle. In both horizontal and vertical direction, extension is constrained to the borders of the low-resolution image.

$$i_{\odot} = \max\left(\frac{i_{ap}}{2}, \min\left(i_{max} - \frac{i_{ap}}{2}, \frac{i_{ul} + i_{lr}}{2}\right)\right) \tag{6.18}$$

$$j_{\odot} = \max\left(\frac{j_{ap}}{2}, \min\left(j_{max} - \frac{j_{ap}}{2}, \frac{j_{ul} + j_{lr}}{2}\right)\right) \tag{6.19}$$

where ($i_{\odot}$,$j_{\odot}$) are the centre coordinates of the extended candidate regions inside the low-resolution image of size (i_{max},j_{max}) and the pixel region corresponding to the aperture angle of the high-resolution sensors (i_{ap},j_{ap}). The extended region $\mathcal{R}_e$ is then calculated using

$$\mathcal{R}_e = \begin{pmatrix} i_{\odot} - \frac{1}{2} i_{ap} \\ j_{\odot} - \frac{1}{2} j_{ap} \\ i_{\odot} + \frac{1}{2} i_{ap} - 1 \\ j_{\odot} + \frac{1}{2} j_{ap} - 1 \\ P(\mathcal{TP}_n) \\ S \\ UC \end{pmatrix} \tag{6.20}$$

6.2.3 Use of Statistically Optimal Regions

In our proposed system up to four statistically optimal regions determined for every road type RT_m are used as candidate regions. These regions are determined in the course of evaluating the candidate region determination process in section 6.2.4 and are given in Tab. 6.3 on p. 156.

6.2.4 Evaluation of Candidate Region Determination

The quality of our determined candidate regions is evaluated using the decision making forms described in section 7.1.2: avoiding (the necessity) to make a decision, choosing a random solution, and using a heuristic candidate region selection approach.

Test Conditions

In order to evaluate the candidate region determination process a set of three test sequences (Trc, Urb, and Mwy) with a total of 1512 frames is used. In these frames, 4748 traffic

participants are labelled manually. As a criterion for evaluation, we use the fraction of all labelled traffic participants entirely inside at least one candidate region of 160×120 px in the original 640×480 px video frame acquired by the fixed camera. This criterion is referred to as *coverage* in the following.

Avoid Decision

If no decision is made during runtime, a static window position is used. The position of this window is decided using statistical frequency of traffic participants' positions in the video frame. In our sensor system, the most frequently occupied region in the vertical direction is $j = [140, 310]$ (cf. Tab 6.2). In the horizontal direction, the variance is significantly higher and dependent upon the road type (cf. Fig. 6.8), which is also pointed out in Torralba *et al.* [146].

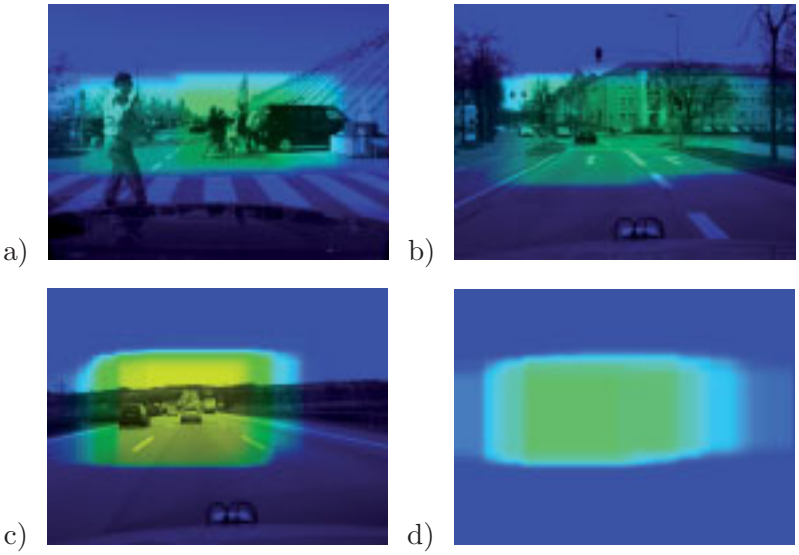


Figure 6.8: Coverage of traffic participants in individual regions. Example images from sequences on a) traffic-calmed road RT_2, b) urban road RT_3, and c) motorway RT_5 are overlayed with the coverage in HSV colour space ranging from 0.0 (blue) to 1.0 (red). Image d) shows the generic coverage considering equal numbers of randomly selected traffic participants from all road types.

Single region The regions with maximum traffic participant coverage for different road types are given in Tab. 6.2. These are also the optimum regions for a single sensor. The coverage of the regions is determine by randomly splitting the 1512 video frames of the test sequences into a training set containing 80% of all video frames and a test set containing 20% of all video frames.

Road type	i_{UL}	j_{UL}	i_{IR}	j_{IR}	Coverage
RT_2	265	175	425	295	0.29
RT_3	260	160	420	280	0.33
RT_5	185	155	345	275	0.58
$RT_{2,3,5}$	145	170	305	290	0.32

Table 6.2: Regions with maximum coverage of traffic participants for different road types.

It can be seen from Tab. 6.2 that the degree of coverage varies for different road types. Less structured environments such as a traffic calmed road (RT_2) and an urban road (RT_3) exhibit a traffic participant coverage of 0.29 and 0.33 respectively. For a highly structured environment such as a motorway (RT_5), a significant coverage of 0.58 is obtained.

The statistically optimal region $RT_{2,3,5}$ given in Tab. 6.2 is also used by the controllable sensors in the case of a resource allocation system failure. This graceful degradation can further be optimised if sensors are able to acquire information about the current road type, e.g. by using our distributed environment model presented in Hermann *et al.* [11]. There, controllable sensors are able to use the statistically optimum region for the current road type given in Tab. 6.2.

Multiple regions If more than one region is observed at the same time, the coverage of traffic participants can be increased. In Tab. 6.3 additional regions are given, providing a substantial increase in coverage

Road type	$N_{\mathcal{R}}$	i_{UL}	j_{UL}	i_{IR}	j_{IR}	Coverage
RT_2	1	265	175	425	295	0.29
	2	105	180	265	300	0.51
	3	400	160	560	280	0.64
	4	0	165	160	285	0.74
RT_3	1	260	160	420	280	0.33
	2	140	155	300	275	0.54
	3	410	170	570	290	0.70
	4	0	175	160	295	0.84
RT_5	1	185	155	345	275	0.58
	2	65	175	225	295	0.76
	3	305	135	465	255	0.88
	4	0	180	160	300	0.91
$RT_{2,3,5}$	1	145	170	305	290	0.32
	2	275	175	435	295	0.58
	3	0	175	160	295	0.68
	4	420	160	580	280	0.78

Table 6.3: Coverage of traffic participants for increasing $N_{\mathcal{R}}$ of statistically optimal regions.

Random Decision

In a random decision approach, a set of $N_{\mathcal{R}}$ randomly positioned candidate regions is selected. We use a uniform distribution vertically constrained to

$$\mathcal{R} = (i_{UL}, j_{UL},\ i_{IR}, j_{IR}) = (0, 85,\ 640, 320)$$

which corresponds to the minimum and maximum vertical values in Tab. 6.3. In Tab. 6.4 the coverage of traffic participants using a random observation pattern for every sensor is given.

	Coverage for			
Road type	$N_{\mathcal{R}} = 1$	$N_{\mathcal{R}} = 2$	$N_{\mathcal{R}} = 3$	$N_{\mathcal{R}} = 4$
RT_2	0.02	0.05	0.07	0.10
RT_3	0.03	0.07	0.11	0.13
RT_5	0.02	0.04	0.06	0.09
$RT_{2,3,5}$	0.02	0.05	0.07	0.09

Table 6.4: Mean coverage of traffic participants for different road types and different numbers of randomly determined regions after 4,000 test runs for every $N_{\mathcal{R}}$. Standard error of the mean is $SE_x < 0.005$ for all values.

The small coverage values in Tab. 6.4 suggest that a random observation pattern is not a viable approach for sensor resource allocation.

Proposed Region Selection Method

In our proposed system, a combination of statistically optimal regions based upon the current road type (section 5.1), traffic participant detection information (section 5.2), and saliency information (section 5.4) is used. In the following, traffic participant coverages using either saliency information, or traffic participant detection information. The combination of all information sources is evaluated in section 6.2.5.

Saliency based Region Selection Using saliency information as discussed in section 6.2.1, the traffic participant coverage for different road types and increasing numbers of observed regions $N_{\mathcal{R}}$ is evaluated (cf. Tab. 6.5).

The coverage values for regions determined using only saliency information in Tab. 6.5 show a significant increase if more regions are observed. Also, traffic participant coverage values for $N_{\mathcal{R}} \geq 3$ using either Walker *et al.* saliency, or Frintrop *et al.* saliency are considerable.

Road type	$N_{\mathcal{R}}$	Coverage for Walker *et al.*	Itti *et al.*	Frintrop *et al.*
RT_2	1	0.13	0.07	0.07
	2	0.33	0.13	0.21
	3	0.43	0.26	0.37
	4	0.48	0.35	0.54
RT_3	1	0.16	0.05	0.08
	2	0.32	0.10	0.20
	3	0.40	0.22	0.37
	4	0.53	0.31	0.49
RT_5	1	0.36	0.25	0.35
	2	0.59	0.49	0.59
	3	0.73	0.62	0.70
	4	0.80	0.73	0.76

Table 6.5: Coverage of traffic participants for increasing numbers of observed regions $N_{\mathcal{R}}$ determined using only saliency information. As saliency detector the methods proposed by Walker *et al.* [118], Itti *et al.* [99], and Frintrop *et al.* [102] are used.

As opposed to our saliency detector evaluation results in Tab. 5.8 where the correlation with a ground truth map is shown to be highest using Frintrop *et al.* saliency, the results in Tab. 6.5 show the actual coverages for candidate regions, which is more relevant for our system. There, the use of Walker *et al.* saliency shows the best overall coverage.

Traffic Participant Detection based Region Selection Using traffic participant detection information as discussed in section 6.2.2, the traffic participant coverage is evaluated for different road types (cf. Tab. 6.6).

	Extended regions		Merged regions	
Road type	$\overline{N}_{\mathcal{R}}$	Coverage	$\overline{N}_{\mathcal{R}}$	coverage
RT_2	1.80	0.47	1.65	0.45
RT_3	0.72	0.50	0.57	0.50
RT_5	1.47	0.61	1.29	0.59

Table 6.6: Coverage of traffic participants using only traffic participant detection information. The number of regions $N_{\mathcal{R}}$ per video frame is dependent upon the detection cascade's results, therefore the mean number of regions $\overline{N}_{\mathcal{R}}$ for both extended regions and merged regions is given.

The coverage values using detected traffic participants in Tab. 6.6 are in the range of 0.45 to 0.61, which is less than using $N_{\mathcal{R}} = 4$ for both static regions (Tab. 6.3) and salient regions (Tab. 6.5). The coverage values must therefore be considered in relation to the small mean number of regions $\overline{N}_{\mathcal{R}} = [0.57, 1.80]$. A comparison of the different candidate region determination strategies is presented in the following.

6.2.5 Comparison of Strategies for Candidate Region Determination

The results presented for statistically optimal regions, salient regions, and regions determined using traffic participant information in section 6.2.4 allow a comparison of overall coverage and region efficiency.

Overall Coverage

First, a diagram of the overall coverage values for the above methods is given in Fig. 6.9. A high coverage indicates that a large fraction of traffic participants is included in the considered regions, which is desirable. The coverage values in Fig. 6.9 are extracted from Tab. 6.3 and Tab. 6.5 for $N_{\mathcal{R}}$=4, and for $\overline{N}_{\mathcal{R}} = [0.57, 1.65]$ in Tab. 6.6 respectively.

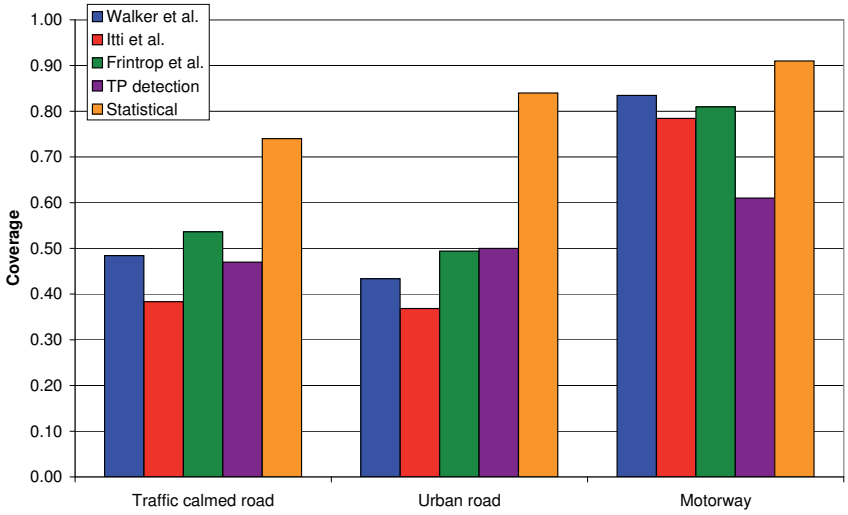


Figure 6.9: Comparison of traffic participant coverages using $N_{\mathcal{R}}$=4 for statistical regions, salient regions, and $\overline{N}_{\mathcal{R}}$ merged traffic participant detection regions.

From the comparison of traffic participant coverages in Fig. 6.9 it can be seen that the statistically determined candidate regions show the best overall coverage, as is implied by statistical optimality. Salient regions show to be dependent upon the environmental situation, with good results in highly structured traffic environments such as a motorway, but poor results in relatively unstructured environments such as urban traffic. The regions determined using traffic participant detection show a coverage in the range $[0.47, 0.61]$, increasing slightly with the degree of structuredness of the road traffic situation.

When examining the computational costs every candidate region requires, it is clear that statistical regions do not require any substantial processing time during runtime as they constitute prior knowledge. At the same time, the overall coverage is statistically

optimal. This view however disregards three main disadvantages of statistical candidate regions:

- Traffic participants inside the statistically optimal regions are less likely to be overlooked by a human driver.
- Traffic participants outside the candidate regions are ignored systematically.
- Information about detected traffic participants is not available.

The first problem is a specific inference when considering driver assistance systems that provide information complementary to the driver's perception. In contrast to this, the second and third problem have an immediate influence on the proposed system's performance. Both saliency information and detected traffic participants help to alleviate the disadvantages of using statistically optimal regions.

Candidate Region Efficiency

Besides overall coverage values, the individual candidate region efficiency must be considered. We define candidate region efficiency to be the ratio of traffic participant coverage per candidate region. The efficiency of the candidate region determination methods compared in Fig. 6.9 can be seen in Fig. 6.10.

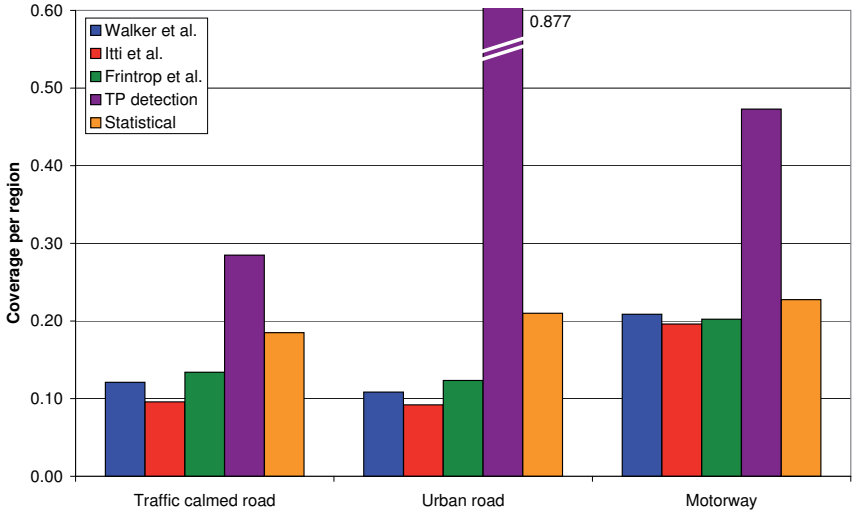


Figure 6.10: Comparison of candidate region efficiency (traffic participant coverage per candidate region) considering $N_{\mathcal{R}}$=4 for statistical regions, salient regions, and $\overline{N}_{\mathcal{R}}$ merged traffic participant detection regions.

The comparison of candidate region efficiencies of the discussed methods in Fig. 6.10 shows that regions determined using traffic participant detection information are most

efficient for all road types, followed by statistically determined candidate regions. As in Fig. 6.10 salient regions are largely dependent upon the complexity of the road traffic environment.

Combined Candidate Region Determination

As a last evaluation step, all candidate regions are subsumed in a combined candidate region set. For the evaluation of candidate region performance, a number of $N_{\mathcal{R}}$ candidate regions is selected from the combined set using two methods. First, a simple combinatorial search determines the region subset with the highest coverage for the given $N_{\mathcal{R}}$. This requires knowledge of ground truth information and therefore yields the maximum possible overall coverage for a given $N_{\mathcal{R}}$ (cf. Fig. 6.11, solid). Second, random selection is used for the region subset. The overall coverage for random selection is also given in Fig. 6.11 (dashed).

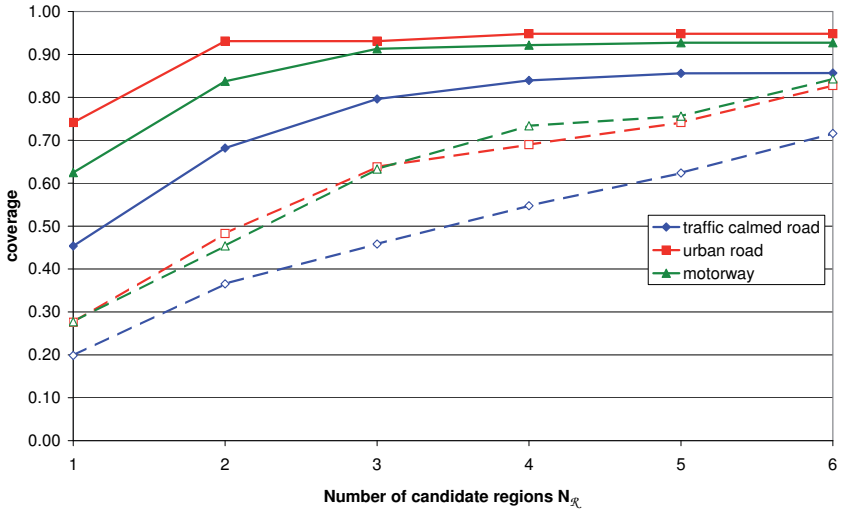


Figure 6.11: Overall coverage for different road types and $N_{\mathcal{R}}$ determining the optimum subset (solid) using ground truth information and a random subset (dashed).

The coverage graphs in Fig. 6.11 show two interesting properties. First, the coverage using a random subset increases approximately linearly with $N_{\mathcal{R}}$. Second, maximum coverage can be attained using three or four candidate regions. This property is highly relevant, as the number of controllable sensor resources in our system is in the same range.

6.3 Discussion of Reasoning Level Modules

In this chapter, the acquisition and fusion of knowledge about the current context and the determination of candidate regions is presented. Below, the determination of traffic participant probability and the candidate region quality measure used for evaluation is discussed.

6.3.1 Traffic Participant Probability Determination

In our proposed system's reasoning level, the individual probability of at least one traffic participant of a certain type to exist inside a candidate region $\mathcal{R}_n$ and the whole environment $\mathcal{R}_\forall$ is determined. The use of a binomial probability

$$P(\mathcal{TP}_n|N \cdot C) = 1 - [1 - P(\mathcal{TP}_n|C)]^N,\ N \geq 1$$

where N is the number of positive detector or classifier results C has its foundation in moral theory discussed in section 2.3.1. There, negotiating the life of a single person against the life and physical integrity of two or more people is considered both unlawful and unethical. As a consequence the probability determined in our reasoning level must be the probability of at least one traffic participant of a certain type to exist, complemented by the probability of no traffic participant of a certain type to exist.

The covariance union information fusion method used in the proposed system relies on the knowledge of the variance of both statistical probabilities $var\left(P(\mathcal{TP}|\mathcal{RT}_m)\right)$ and dynamic probabilities $var\left(P_k(\mathcal{TP}|C)\right)$. In our proposed system, these variances are assumed to be

$$var\left(P(\mathcal{TP}|\mathcal{RT}_m)\right) = var\left(P_k(\mathcal{TP}|C)\right) = 0.2^2 \cdot I_5$$

A quantitative evaluation of the variances therefore presents future work. While the variance of statistical probabilities on different road can be obtained from the same statistical sources, the variance of dynamic probabilities depends on the implemented system and can be evaluated using ground truth information.

Apart from determining the fused traffic participant probabilities $P_k^{\cup}(\mathcal{TP})$, the variance $var\left(P_k^{\cup}(\mathcal{TP})\right)$ of the fusion result is also known. This variance can be used as an indicator of the fused probability values' validity. As a consequence, a probability with a small

variance can be assigned a higher weight than a probability with larger variance as an extension to our proposed decision making process.

6.3.2 Candidate Region Quality

For our evaluation of the candidate region determination methods in section 6.2.4 it must be noted that the optimum coverage values are calculated using ground truth information. This is valid insofar as a traffic participant not included in the candidate region cannot be classified. However, traffic participants that are not recognised by our trained classifier cascades fail to provide additional information. Therefore the evaluated coverage values present the theoretically maximal coverage of our system.

In chapter 7, a multiobjective resource allocation concept is proposed, where traffic participant coverage is only one objective amongst others, such as TTC and the region's present uncertainty UC.

Chapter 7

Contextual Resource Allocation

In the highest level of abstraction, contextual knowledge provided by the reasoning level is analysed to efficiently allocate sensor resources in the present environment.

This chapter is organised as follows. In section 7.1 the nature of the resource allocation problem and our proposed resource allocation concept is investigated. The determination of combined utility is discussed in section 7.2, followed by a discussion and evaluation of both sensor resource allocation heuristics in section 7.3, and computational resource allocation heuristics in section 7.4. Our contextual resource allocation method is evaluated in section 7.5 and discussed in section 7.6.

7.1 Resource Allocation Concept

In this section, the nature of the resource allocation problem and our proposed resource allocation concept is investigated. In section 7.1.1 the influences on the decision making process are identified. Different forms of decision making are presented in section 7.1.2. The resource allocation problem is formalised in section 7.1.3, and our proposed resource allocation concept presented in section 7.1.4.

In Fig. 7.1, an overview of the proposed resource allocation process is given. It can be seen that the allocation process is partitioned into two sequential steps. First, sensor-region combinations $\{\mathcal{S}, \mathcal{R}\}$ are determined by a sensor resource allocation module. The resulting partial allocations $\mathcal{A}_{\mathcal{S}_n}$ are allocated a set of classifiers $\mathcal{C}_{I\!P,n}$ and a classifier priority $\mathcal{P}_n$ towards an allocation $\mathcal{A} = \{\mathcal{S}, \mathcal{R}, \mathcal{C}, \mathcal{P}\}$.

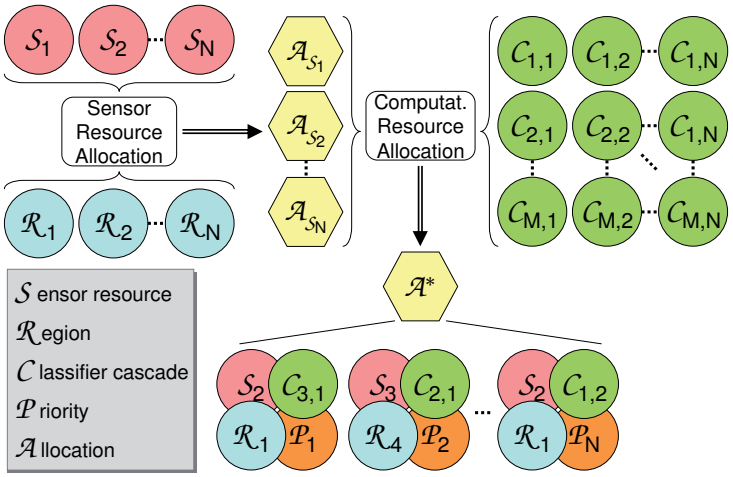


Figure 7.1: Overview of the resource allocation process. First, sensor-region combinations are determined by a sensor resource allocation module. The resulting partial allocations $\mathcal{A}_{\mathcal{S}_n}$ are allocated a set of classifiers $\mathcal{C}_{\mathcal{IP},n}$ and a classifier priority $\mathcal{P}_n$. The optimal allocation $\mathcal{A}^*$ then represents the set of $\{\mathcal{S}, \mathcal{R}, \mathcal{C}, \mathcal{P}\}$ tuples with the highest estimated utility.

7.1.1 Influences on Decision Making

A decision making process is influenced by a number of factors which can be subsumed into the five fields below:

- nature of the environment,
- existing information,
- predefined objectives,
- decision making strategy, and
- computational complexity.

In the following, the above fields are described and the reinforcing relationship between resource allocation and information acquisition is pointed out.

Nature of the Environment

Every decision depends on the environment it is a part of and on the environment it acts upon by making a decision. Our environment is formally described using an ontology given in Fig. 1.2 on p. 4. The environment the decision making process can act upon consists of both sensor resources and computational resources.

Existing Information

Two types of information exist: prior information and dynamic information. While prior information is available from the first moment of system operation, dynamic information is be acquired using sensors and interpreted by high-level data processing methods.

In our proposed system, prior information covers information such as a digital road map, traffic accident statistics, sensor properties, and available computational resources. Dynamic information acquired by the sensors is available at the system's sensor level described in chapter 3 and processed towards a reasoning level representation presented in chapter 6.

Predefined Objectives

Predefined objectives are used in the decision making process to evaluate the quality of different candidate solutions. This evaluation can either result in a boolean evaluation of the candidate solution's feasibility or can result in a numeric evaluation of its fulfilment of one or more objectives. In our system five objectives Ω_n to be maximised are defined and motivated in section 7.2.2.

Decision Making Strategy

A central factor of the decision making process is the strategy used to determine, evaluate, and select candidate solutions. Decision making methods proposed in the literature are discussed in sections 2.3 and 2.4. The candidate region determination is described in section 6.2. The evaluation and selection of candidate regions is presented and evaluated in sections 7.2 to 7.5.

Computational Complexity

In our proposed system, computational resources are an important factor for decision making processes in two respects. First, the decision making process has to take the computational requirements of candidate solutions into account. Second, the decision making process itself must not be computationally expensive as this delays the observation of a candidate region and deprives other processes of computational resources in a single processor environment.

Reinforcing Relationship of Resource Allocation

Considering the present problem of allocating both sensor resources and computational resources, our decision making process is in a reinforcing relationship with the process of information acquisition as illustrated in Fig. 7.2.

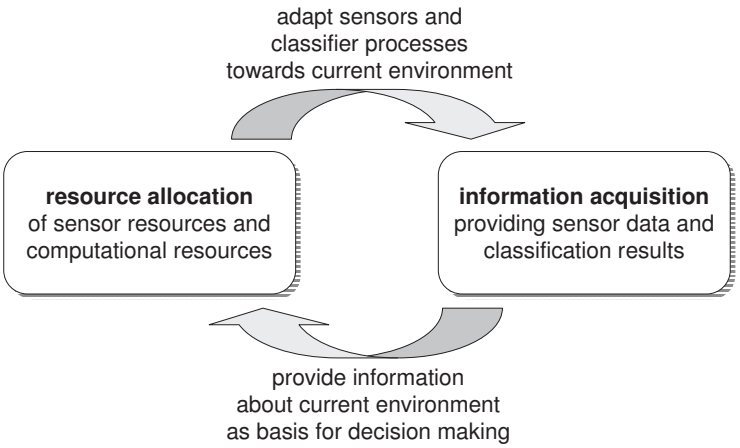


Figure 7.2: Reinforcing relationship of resource allocation and information acquisition.

The reinforcing relationship between information acquisition and resource allocation in Fig. 7.2 results in two specific properties of our proposed system. First, a good performance of either component leads towards an upwards spiral in quality for both components, yet the opposite is also true for a poor performance of either component. It is therefore necessary to ensure that both components perform well. Second, the system requires prior information as a basis for initial operation. Otherwise resource allocation has to rely on dynamical information about the environment, which in turn suffers from a suboptimal resource allocation.

7.1.2 Forms of Decision Making

Decision making can take various forms depending on the influences described in section 7.1.1. A set of commonly used techniques for decision making in order of ascending complexity is identified in the enumeration below.

1. avoid (necessity) to make decisions *or* delegate decisions to another instance,
2. choose random solution,
3. evaluate selected solutions, choose best solution, and
4. evaluate all solutions, choose best solution.

The individual items of the enumeration are discussed below. As an illustrative example the control of a PTZ camera in a car is considered.

Decision making avoidance

Decision making avoidance is not understood in the form of procrastination, but describes a linear process design. In a linear process, no decisions have to be made. In our example, this can be a fixed camera continuously observing the same field of view. In this system, the decision about the field of view is made once during installation, whereas during operation no further decision upon the field of view is possible or necessary.

Delegation of Decision Making

Delegation of the decision making process towards another instance is a special case of decision making avoidance. It requires a means of communication but no decision is made inside the system itself. The system can delegate the control over the field of view towards the driver or rather an external instance tracking the driver's gaze direction. The external instance can then either choose the field of view the driver is currently observing (using the driver's attention and experience in road traffic) or observe the part of the environment the driver is currently neglecting (so as to compensate for the driver's inattention towards that region). In either case, no decision making is necessary inside the resource allocation system.

Random Decision Making

The first strategy that involves making a decision inside the system is to make a random decision among a set of candidate solutions. This does not require any information about the current environment or situation, as it is an entirely autonomous system. Considering a PTZ camera, this causes the field of view to change arbitrarily, ideally providing an equally distributed, fair allocation of sensor resources towards all regions.

Evaluation of selected Solutions

The quality of evaluating a subset of all possible solutions depends upon the quality of the used search heuristic. A search heuristic requires knowledge about the current environment, predefined objectives, and a method to evaluate whether a solution is practicable

or not. A heuristic method evaluates selected candidate solutions until a termination condition is fulfilled. Examples for predefined termination conditions are the number of iterations, a solution quality indicator, or a timer. After termination, the solution evaluated best so far is considered to be the globally optimal solution. Using a heuristic approach for a PTZ camera requires to evaluate a limited set of regions determined by a search heuristic, choosing the candidate solution exhibiting the best estimated utility.

Evaluation of all Solutions

An alternative to evaluating solutions until a certain termination condition is fulfilled is to evaluate all solutions. This method ensures that the globally best solution is found. Apart from the decision whether the objectives' minimum criteria are met, a decision has to be made which solution from the set of practicable solution is the best solution. For a controllable camera in a car this results in focusing the region in the environment with the highest expected utility.

7.1.3 Formalisation of Resource Allocation Problem

A formalisation scheme for resource allocation problems is proposed in Chevaleyre et al. [83]. The scheme allows determination of the nature of the resource type to be allocated, the preference representation and a common utility measure. Resources can be classified using a set of properties dependent both on the type of resource itself and on desired allocation properties.

- Resources can be continuous or discrete.
- Resources can be shareable or non-shareable.
- Resources can be static or volatile within the timespan a resource is allocated.
- Resources can be unique or consist of multiple identical units.

In the presented system both sensors and classifier processes are considered resources. These resources can be allocated towards regions and sensor data respectively. The resources' properties are listed in Tab. 7.1 and are described in more detail below.

It can be seen in Tab. 7.1 that the properties of sensors and classifier processes are mostly identical. First, both sensors and classifier processes are discrete resources, as no fraction of each resource can be allocated. This class of problems, having a finite set of

Property	Sensors	Classifier process
Divisibility	discrete ($\mathbb{N}$)	discrete ($\mathbb{N}$)
Shareability	non-shareable	non-shareable
Variability	quasi-static	static
Exchangeability	unique	multi-unit
Allocation instance	Region	Sensor data

Table 7.1: Characterisation of sensor allocation problem and classifier process allocation problem using basic properties.

possible feasible solutions, is also understood as a combinatorial problem by Ehrgott [85]. Second, resources are not shareable as every sensor can only observe a single region and every classifier process can only examine a single set of sensor data. Third, a sensor's performance can change over time due to adverse environment conditions (e.g. fog, soiling, or vibrations) but can be considered static within the timespan of a single allocation. Classifier processes are considered static resources. Fourth, in our proposed system, sensors are unique, but the same classifier process can be assigned different sensor-region combinations. For sensors this changes if identical sensors with overlapping fields of view are installed.

The main difference between sensors and classifier processes is that sensors $\mathcal{S}$ are allocated towards regions $\mathcal{R}$, whereas classifier processes $\mathcal{C}$ are allocated towards the sensor data acquired by a sensor-region combination, which is defined as a partial allocation $\mathcal{A}_{\mathcal{S}_n}$ in section 7.1.4 below.

7.1.4 Proposed Resource Allocation Concept

A partial allocation $\mathcal{A}_{\mathcal{S}_n}$ for a given sensor $\mathcal{S}_n$ contains the region $\mathcal{R}_{\mathcal{S}_n}$ observed by the sensor, a classifier process $\mathcal{C}_m$ used on the sensor data.

$$\mathcal{A}_{\mathcal{S}_n} = \{\mathcal{S}_n,\ \mathcal{R}_{\mathcal{S}_n},\ \mathcal{C}_m\} \tag{7.1}$$

If more than one classifier process $\mathcal{C}_m$ is allocated to a sensor-region combination, $\mathcal{A}_{\mathcal{S}_n}$ is extended by the priority $\mathcal{P}_{n,m}$ of the allocated classifier processes.

$$\mathcal{A}_{\mathcal{S}_n} = \left\{ \begin{array}{cccc} \mathcal{S}_n, & \mathcal{R}_{\mathcal{S}_n}, & \mathcal{C}_1, & \mathcal{P}_{n,1} \\ \mathcal{S}_n, & \mathcal{R}_{\mathcal{S}_n}, & \mathcal{C}_2, & \mathcal{P}_{n,2} \\ & & \vdots & \\ \mathcal{S}_n, & \mathcal{R}_{\mathcal{S}_n}, & \mathcal{C}_m, & \mathcal{P}_{n,m} \end{array} \right\} \tag{7.2}$$

A complete allocation $\mathcal{A}$ then consists of $N_{\mathcal{S}}$ allocated sensors, with associated regions, classifiers, and classifier priorities.

$$\mathcal{A} = \left\{ \begin{array}{cccc} \mathcal{S}_1, & \mathcal{R}_{\mathcal{S}_n}, & \mathcal{C}_1, & \mathcal{P}_{n,1} \\ \mathcal{S}_1, & \mathcal{R}_{\mathcal{S}_n}, & \mathcal{C}_2, & \mathcal{P}_{n,2} \\ & & \vdots & \\ \mathcal{S}_{N_{\mathcal{S}}}, & \mathcal{R}_{\mathcal{S}_{N_{\mathcal{S}}}}, & \mathcal{C}_m, & \mathcal{P}_{N_{\mathcal{S}},m} \end{array} \right\} \tag{7.3}$$

An allocation is assumed to be optimal ($\mathcal{A}^*$) if the combined overall utility for an allocation $\mathcal{U}(\mathcal{A})$ as defined in section 2.3.3 becomes maximal

$$\mathcal{A}^* = \arg\max_{\mathcal{A}} \ \mathcal{U}(\mathcal{A}) \tag{7.4}$$

The proposed resource allocation concept is also illustrated in Fig. 7.1.

Computational Complexity

The presented resource allocation is an optimisation problem dependent upon the number of candidate regions $N_{\mathcal{R}}$, the number of sensors $N_{\mathcal{S}}$, the number of traffic participant types $N_{\mathbb{TP}}$, and the number of classifier processes trained for every traffic participant type $N_{\mathcal{C}/\mathbb{TP}}$. The resulting computational complexity is given in Eq. 7.5.

$$O\left((N_{\mathcal{R}})^{N_{\mathcal{S}}} \cdot (N_{\mathcal{C}/\mathbb{TP}})^{N_{\mathbb{TP}}} \cdot N_{\mathbb{TP}}!\right) \tag{7.5}$$

where $(N_{\mathcal{R}})^{N_{\mathcal{S}}}$ is the number of possible sensor-region combinations, $(N_{\mathcal{C}/\mathbb{TP}})^{N_{\mathbb{TP}}}$ the number of possible classifier combinations, and $N_{\mathbb{TP}}!$ the number of possible classifier prioritisations. For our proposed system the ranges of complexity-relevant numbers used in our evaluation process are given in Tab. 7.2.

		Minimum	Maximum
(Virtual) sensors	$N_{\mathcal{S}}$	2	7
Candidate regions	$N_{\mathcal{R}}$	5	15
Traffic participant types	$N_{\mathbb{TP}}$	3	5
Classifier scalings	$N_{\mathcal{C}/\mathbb{TP}}$	1	3
Possible allocations	$N_{\mathcal{A}}$	150	$2.6{\cdot}10^{12}$

Table 7.2: Range of complexity-relevant numbers $N_{\mathcal{S}}$, $N_{\mathcal{R}}$, $N_{\mathbb{TP}}$, and $N_{\mathcal{C}/\mathbb{TP}}$ and number of possible allocations for our proposed system.

It can be seen from Tab. 7.2, that an exhaustive evaluation of all possible allocations $\mathcal{A}$ quickly becomes infeasible. It can also be seen that for our proposed system, the number of possible sensor-region combinations is the governing factor for our system's complexity. Considering the actual numbers used for our system evaluation in section 7.5, a total of

$$N_{\mathcal{A}} = \left((10)^3 \cdot (3)^3 \cdot 3!\right) = \left(10^3 \cdot 9 \cdot 6\right) = 5.4 \cdot 10^4$$

different allocations exist.

The same is not true if the complex resource allocation problem is split into a set of sub-problems, each of which can be solved efficiently by considering a single problem domain. Using a classical *divide and conquer* approach as defined in Cormen *et al.* [192] is problematic as this requires the independence of all sub-problems. For our given problem the choice of $\mathcal{C}$ is dependent on $\mathcal{S}$ and $\mathcal{R}$ and the process priority $\mathcal{P}$ dependent on the set of classifier processes $\mathcal{C}$.

Partition of the Resource Allocation Process

We propose a partition of the resource allocation process into two sequential steps: sensor resource allocation determining optimal sensor-region combinations and computational resource allocation performing classifier allocation and prioritisation as illustrated in Fig. 7.1.

The division into two sequential modules helps to reduce the complexity of the individual modules to a base complexity of $O\left((N_{\mathcal{R}})^{N_{\mathcal{S}}}\right)$ for sensor resource allocation and a computational resource allocation with a base complexity of $O\left((N_{\mathcal{C}/\mathcal{IP}})^{N_{\mathcal{IP}}} \cdot N_{\mathcal{IP}}!\right)$ for classifier allocation and prioritisation. As these complexities are hard to manage under real-time constraints, efficient search heuristics for both sensor resource allocation in section 7.3 and computational resource allocations in section 7.4 are presented. Both methods require a measure to evaluate the quality of individual candidate solutions. In our system this quality is expressed using the notion of combined utility, which is discussed in section 7.2 below.

7.2 Determination of Combined Utility

Our proposed system uses a combined utility concept as an internal measure for the quality of candidate solutions. In this section, the determination of combined utility is

demonstrated using an example road traffic scene from our motorway sequence MWY.

7.2.1 Introduction of Example Scene

In Fig. 7.3 an example video frame is labelled with traffic participants detected on a low-resolution video image. Candidate regions which are close and small enough are combined. In Fig. 7.3 this is the case for $\mathcal{R}_{23}$.

Figure 7.3: Example video frame labelled with detected traffic participants (human: red, vehicle: green). The detected human traffic participant in $\mathcal{R}_5$ is a false positive. Candidate regions which are close and small enough are combined ($\mathcal{R}_{23}$, blue).

In Fig. 7.4 candidate regions from Fig. 7.3 are extended. Additionally, the region with highest statistical traffic participant coverage for RT_5 from Tab. 6.2 is drawn into Fig. 7.4 ($\mathcal{R}_6$, violet), as well as the most salient region ($\mathcal{R}_7$, orange).

Figure 7.4: Candidate regions from Fig. 7.3 are extended. Note that $\mathcal{R}_5$ remains at its original size, since it is too large to be observed by high-resolution sensors.

The candidate regions are then transferred on the saliency map of the video image given in Fig. 7.5.

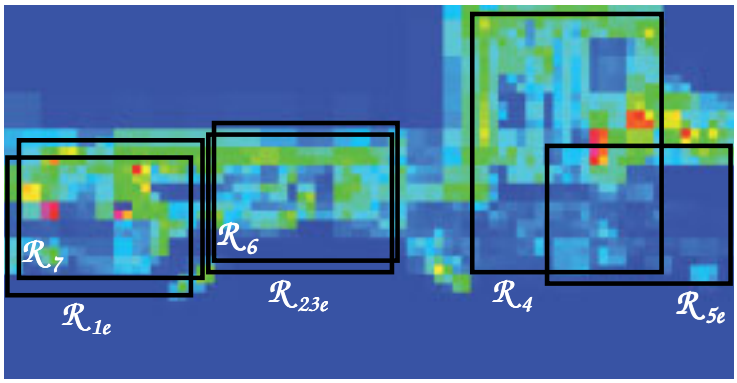


Figure 7.5: Extended candidate regions from Fig. 7.4 are transferred to the saliency map.

From the regions shown in Fig. 7.4, a list of candidate regions, information about detected objects and the regions' mean bottom-up saliency values $\overline{S}$ are given in Tab. 7.3.

Region	width [px]	height [px]	N_H	N_V	$\overline{S}$
$\mathcal{R}_{1e}$	160	120	0	1	46.9
$\mathcal{R}_{23e}$	160	120	0	2	36.8
$\mathcal{R}_4$	170	230	0	1	44.1
$\mathcal{R}_{5e}$	160	120	1	0	42.4
$\mathcal{R}_6$	160	120	0	2	38.6
$\mathcal{R}_7$	160	120	0	1	53.3

Table 7.3: Example regions from Fig. 7.4 with respective widths, heights, number of detected traffic participants (human $\mathcal{TP}_H$ or vehicle $\mathcal{TP}_V$), and mean saliency values $\overline{S}$.

Using the traffic participant probability $P(\mathcal{TP}_n|\mathcal{RT}_m)$ for the current road type from Tab. 6.1 the traffic participant probabilities are obtained.

$$P(\mathcal{TP}_H|\mathcal{RT}_5) = P(\mathcal{TP}_1|\mathcal{RT}_5) + P(\mathcal{TP}_2|\mathcal{RT}_5) + P(\mathcal{TP}_3|\mathcal{RT}_5) = 0.032$$

$$P(\mathcal{TP}_V|\mathcal{RT}_5) = P(\mathcal{TP}_1|\mathcal{RT}_4) + P(\mathcal{TP}_5|\mathcal{RT}_5) = 0.968$$

Traffic participant probabilities considering detection results are calculated using Eq. 6.1 to 6.6 assuming the probability for human traffic participants $P(\mathcal{TP}_H|\mathcal{RT}_5)$ and vehicles $P(\mathcal{TP}_V|\mathcal{RT}_5)$ as prior probabilities in absence of a detection history for the given example. The resulting traffic participant probabilities are given in Tab. 7.4.

$\mathcal{TP}_n$	$P(\mathcal{TP}_n\|R\mathcal{T}_m)$	$P(C\|\mathcal{TP}_n)$	$P(\mathcal{TP}_n\|C)$	$P(\mathcal{TP}_n\|2C)$	$P(\mathcal{TP}_n\|\neg C)$
$\mathcal{TP}_H$	0.032	0.7	0.104	0.197	0.012
$\mathcal{TP}_V$	0.968	0.8	0.992	0.999	0.883

Table 7.4: Probabilities for traffic participant groups for one positive $P(\mathcal{TP}_n|C)$, two positives $P(\mathcal{TP}_n|2C)$, and negative $P(\mathcal{TP}_n|\neg C)$ detection results, assuming $P(C|\neg\mathcal{TP}_n) = 0.20$.

The posterior traffic participant probabilities $P(\mathcal{TP}_H)$ and $P(\mathcal{TP}_V)$ from Tab. 7.4 are decomposed into individual traffic participant probabilities using Eq. 6.9 to 6.12. The resulting traffic participant type probabilities $P(\mathcal{TP}_{1,4,5})$ are given in Tab. 7.5.

Region	N_H	N_V	$P(\mathcal{TP}_H)$	$P(\mathcal{TP}_V)$	$P(\mathcal{TP}_1)$	$P(\mathcal{TP}_4)$	$P(\mathcal{TP}_5)$
$\mathcal{R}_{1e}$	0	1	0.012	0.992	0.002	0.750	0.242
$\mathcal{R}_{23e}$	0	2	0.012	0.999	0.002	0.755	0.244
$\mathcal{R}_4$	0	1	0.012	0.992	0.002	0.750	0.242
$\mathcal{R}_{5e}$	1	0	0.104	0.883	0.017	0.668	0.215
$\mathcal{R}_6$	0	2	0.012	0.999	0.002	0.755	0.244
$\mathcal{R}_7$	0	1	0.012	0.992	0.002	0.750	0.242

Table 7.5: Decomposed traffic participant probability for example regions. Probabilities for individual traffic participant types are decomposed from $P(\mathcal{TP}_H)$ and $P(\mathcal{TP}_V)$ using the road type dependent traffic participant distributions.

The traffic participant probabilities from Tab. 7.5 are fused with the statistical traffic participant type probabilities using the covariance union method presented in section 6.1.3. The resulting fused probabilities using the covariance union method are given in Tab. 7.6.

Region	N_H	N_V	$P^{\cup}(\mathcal{TP}_1)$	$P^{\cup}(\mathcal{TP}_4)$	$P^{\cup}(\mathcal{TP}_5)$
$\mathcal{R}_{1e}$	0	1	0.003	0.741	0.239
$\mathcal{R}_{23e}$	0	2	0.003	0.743	0.240
$\mathcal{R}_4$	0	1	0.003	0.741	0.239
$\mathcal{R}_{5e}$	1	0	0.011	0.700	0.226
$\mathcal{R}_6$	0	2	0.003	0.743	0.240
$\mathcal{R}_7$	0	1	0.003	0.741	0.239

Table 7.6: Fused traffic participant probability for example regions using the covariance union method.

7.2.2 Objectives for Utility Optimisation

In order to evaluate the quality of different allocations, a set of objectives must be defined. In section 1.2.2 the goal of our active vision system is defined as

- protect the passengers of the ego-vehicle and other traffic participants by reducing uncertainty about traffic participants with whom a collision is possible.

This goal is decomposed into a set of five mutually independent objectives Ω_n to be maximised:

- regions with vulnerable or dangerous traffic-participants Ω_1,
- salient regions Ω_2,
- regions with critical time-to-collision values Ω_3,
- regions for which observation results in a high uncertainty reduction Ω_4, and
- regions that can be observed and processed in the available time Ω_5.

Below each objective is discussed and the regions' utilities $\mathcal{U}_n(\mathcal{R}_m)$ using objective Ω_n for our example road traffic scene introduced in section 7.2 are determined.

Prioritisation of Vulnerable and Dangerous Traffic Participants

Apart from observing the frequency of traffic accidents given in Tab. 6.1, the degree of suffered injuries $\mathcal{I}$ of these is of importance. The conditional probability $P(\mathcal{I} \mid \mathcal{TP}_n, \mathcal{RT}_m)$ of injuries for road traffic accidents with injured persons differentiated by the type of road traffic participation $\mathcal{TP}_n$, and road type $\mathcal{RT}_m$ is shown in Tab. 7.7.

Road type	Participant	Lethal injuries	Severe injuries	Mild injuries
Urban traffic	Pedestrian	0.019	0.329	0.779
	Bicycle	0.006	0.236	0.854
	Motorcycle	0.010	0.271	0.853
	Car	0.005	0.141	1.115
	Lorry	0.012	0.139	1.057
Country road	Pedestrian	0.124	0.486	0.647
	Bicycle	0.031	0.425	0.697
	Motorcycle	0.044	0.493	0.660
	Car	0.034	0.337	1.154
	Lorry	0.034	0.260	1.076
Motorway	Pedestrian	0.391	0.348	0.870
	Bicycle	0.167	0.333	0.500
	Motorcycle	0.061	0.423	0.623
	Car	0.027	0.274	1.315
	Lorry	0.041	0.307	1.115

Table 7.7: Mean number of injuries $\mathcal{I}$ per road traffic accident with injured persons. Mild injuries are injuries that require less than 24 hours of stationary medical treatment, lethal injuries are injuries leading to death within 30 days from the injuries caused by the accident. Source: Federal Statistical Office Germany [186], Tab. UJ 22 (1-3).

With the relative frequency of road traffic accidents and the resulting effects thereof known, a failure mode and effects analysis (FMEA, e.g. Stamatis [193]) can be performed. This is usually done by multiplying the probabilities of a failure with the associated severity of the outcome. However, considering moral theories and legal viewpoints discussed in section 2.3.1 of the literature review, this appears problematic at best.

First, a preference of a single lethal injury as compared to multiple lethal injuries in FMEA is intuitive, but neither legally possible nor morally sound, as life cannot be negotiated against life. Second, the valuation of many non-lethal injuries against a single lethal injury is doubtful for the same reasons. Third, preference of an inevitable single lethal injury ($1 \times 1.00 = 1.00$) over a small probability of lethal injuries for many ($1000 \times 0.0011 = 1.10$) is a frequent example against the use of FMEA (cf. section 2.3.1, Zalta [73]). Still, FMEAs are regularly conducted for safety critical functions, e.g. in automotive, aerospace, and general manufacturing industries [194], for medical devices [195], or software in general [196].

It is pointed out in section 2.3.1 that the raised moral problems do not bear as much weight for our proposed system as it is intended for driver assistance systems with a human driver responsible for all actions, as opposed to an autonomous driving system. However to efficiently provide additional safety, a prioritisation of regions to be observed must be performed in a manner that ensures that risk and possible adverse consequences are minimised.

Utilitarian Severity Valuation A purely utilitarian approach of determining the severity of a possible accident involves the valuation of mild, severe, and lethal injuries. Besides apparent moral problems, the actual severity factors for the individual injuries cannot be determined objectively. As a means of overcoming this dilemma, the socio economic cost of an injury caused by road traffic accidents are frequently considered in utilitarian approaches, using the numbers for 2004 given in Tab. 7.8.

Injuries	Socio economic cost	Severity s
Mild injuries	3,885 Euro	0.04
Severe injuries	87,269 Euro	1.00
Lethal injuries	1,161,885 Euro	13.31

Table 7.8: Possible determination of severity s for mild, severe and lethal injuries caused by road traffic accidents based upon socio economic cost for 2004 given in Höhnscheid and Straube [197].

In order to conduct an FMEA the mean severity $\widehat{s}(\mathcal{TP}_n, \mathcal{RT}_m)$ for a traffic accident with injured people is determined by summing the products of the conditional probability $P(\mathcal{I}|\mathcal{TP}_n, \mathcal{RT}_m)$ for road type and traffic participant with the respective severity $s(\mathcal{I})$ for all levels of injury.

$$\widehat{s}(\mathcal{TP}_n, \mathcal{RT}_m) = \sum_{\mathcal{I}} P(\mathcal{I} \mid \mathcal{TP}_n, \mathcal{RT}_m) \cdot s(\mathcal{I}) \tag{7.6}$$

Using $P(\mathcal{I} \mid \mathcal{TP}_n, \mathcal{RT}_m)$ from Tab. 7.7 and the severity $s(\mathcal{I})$ in Eq. 7.6 the mean severity values $\widehat{s}$ are determined in Tab. 7.9.

Road type	Pedestrian	Bicycle	Motorcycle	Car	Lorry
Urban traffic	0.659	0.350	0.438	0.251	0.347
Country road	2.162	0.862	1.105	0.841	0.756
Motorway	5.596	2.575	1.261	0.689	0.900

Table 7.9: Mean severity $\widehat{s}(\mathcal{TP}_n, \mathcal{RT}_m)$ for a traffic accident differentiated by road type $\mathcal{RT}$ and type of traffic participation.

Road Type Dependent FMEA In order to calculate the relative FMEA severity value for a given road type $s_{rel}(\mathcal{TP}_n, \mathcal{RT}_m)$ the product of the relative frequency of traffic accidents with injuries $P(\mathcal{TP}_n|\mathcal{RT}_m)$ and the mean resulting severity $\widehat{s}(\mathcal{TP}_n, \mathcal{RT}_m)$ thereof are determined.

$$s_{rel}(\mathcal{TP}_n, \mathcal{RT}_m) = \frac{P(\mathcal{TP}_n|\mathcal{RT}_m) \cdot \widehat{s}(\mathcal{TP}_n, \mathcal{RT}_m)}{\sum_{\iota=1,2,..,5} P(\mathcal{TP}_\iota|\mathcal{RT}_m) \cdot \widehat{s}(\mathcal{TP}_\iota, \mathcal{RT}_m)} \tag{7.7}$$

The relative FMEA severity level s_{rel} calculated for different road types using Eq. 7.7 is shown in Fig. 7.6 .

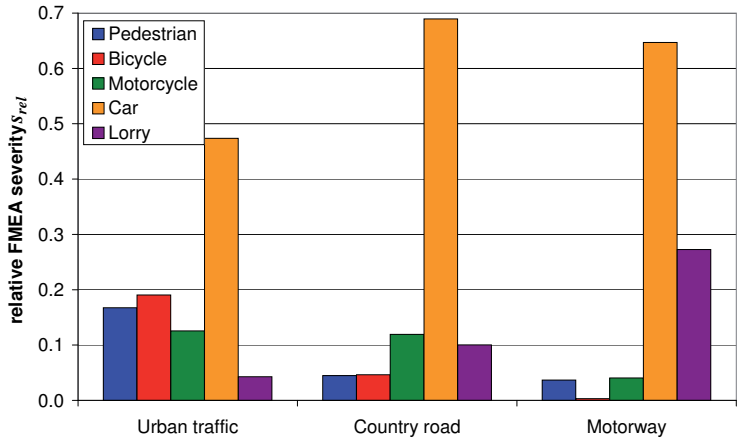


Figure 7.6: Relative FMEA severity level s_{rel} differentiated by $\mathcal{TP}$ and $\mathcal{RT}$ using Eq. 7.7 on the values given in Tab. 6.1 and 7.9.

In Fig. 7.6 the outstanding role of cars in road traffic becomes apparent. Due to the large number of car accidents, the highest road type dependent FMEA score is awarded to cars for every considered road type, including urban traffic.

FMEA based on Traffic Participant Probabilities For prioritisation, the severity $\widehat{s}$ of a traffic accident given in Tab. 7.9 is used to determine the relative observation priority of each candidate region. For the example's road type (motorway, RT_5) the relevant severity scores are $\widehat{s}_1$=5.596, $\widehat{s}_4$=0.689 and $\widehat{s}_5$=0.900.

For every candidate region, priority can be computed as the sum of all severity values per traffic-participant class TP_n multiplied with the fused probability $P_k^{\cup}(\mathit{TP}_n)$ of the respective traffic-participant class.

$$\Omega_1' = \sum_n P_k^{\cup}(\mathit{TP}_n) \cdot \widehat{s}(\mathit{TP}_n, \mathit{RT}_m) \tag{7.8}$$

For the example regions' traffic participant probability distributions given in Tab. 7.6 using Eq. 7.8 results in the objective values in Tab. 7.10.

Region	TP_1	TP_4	TP_5	Ω_1'
$\mathcal{R}_{1e}$	0.017	0.511	0.215	0.742
$\mathcal{R}_{23e}$	0.017	0.512	0.216	0.745
$\mathcal{R}_4$	0.017	0.511	0.215	0.742
$\mathcal{R}_{5e}$	0.062	0.482	0.203	0.747
$\mathcal{R}_6$	0.017	0.512	0.216	0.745
$\mathcal{R}_7$	0.017	0.511	0.215	0.742

Table 7.10: Example regions' severity scores $\widehat{s}$ used as objective Ω_1' calculated as the sum of individual severity scores for traffic participant types $\mathit{TP}_{1,4,5}$.

Unsupervised Saliency

Unsupervised saliency is used as a second objective Ω_2' in our proposed system. For this, the region's mean saliency value $\overline{S}(\mathcal{R}_m)$ given in Tab. 7.3 is used directly (cf. Tab. 7.11). The benefit of prioritising more salient regions is discussed in section 6.2.4 and is especially prominent if a traffic participant is not detected using low-resolution video data.

Region	$\overline{S} = \Omega_2'$
$\mathcal{R}_{1e}$	46.9
$\mathcal{R}_{23e}$	36.8
$\mathcal{R}_4$	44.1
$\mathcal{R}_{5e}$	42.4
$\mathcal{R}_6$	38.6
$\mathcal{R}_7$	53.3

Table 7.11: Example regions' mean saliency values $\overline{S}$ used as objective Ω_2'.

Time-to-Collision

Time-to-collision (TTC) is described as an effective measure to assess the severity of road-traffic conflicts by Van der Horst and Hogema [180]. There, traffic situations with a TTC of less than 1.5 s are considered critical. Present automotive driver assistance systems usually provide a cascade of actions based upon the remaining TTC. From Kühn *et al.* [198] a generic scheme of five phases is derived in Tab. 7.12.

t_{TTC} [ms]	Phase	Example actions
1000-1500	Warning	Visual, acoustic, or haptic warnings
500-1000	Assistance	Autonomous braking
100-500	Pre-crash	Activation of reversible safety systems
10-100	Pre-fire	Belt pre-tensioning
0-10	Pre-set	Parametrisation of airbag system

Table 7.12: Traffic situations differentiated using time-to-collision information.

It can be seen from Tab. 7.12 that, as the car enters the pre-crash phase ($t_{TTC} \leq 500$ ms), the crash cannot be avoided by any action taken. In this phase, an active vision system cannot provide significant information to the vehicle's active safety systems. It is therefore advantageous to discontinue operation in order to leave more bandwidth on the bus system for safety applications.

Knowledge about TTC is used in our systems as an indicator of traffic participant relevance. There, traffic participants with a t_{TTC} =1500 ms are considered to have the highest relevance, whereas traffic participant with a t_{TTC} <500 ms or very large TTC are considered to have a low relevance for our active vision system.

The assignment of utility for objective Ω_3' is then performed using Eq. 7.9

$$\Omega_3' = \begin{cases} e^{-\frac{t_{TCC}-T_{PC}}{T_1}} \cdot \left(1 - e^{-\frac{t_{TCC}-T_{PC}}{T_2}}\right) & \text{if } t_{TC} > T_{PC} \\ 0 & \text{otherwise} \end{cases} \tag{7.9}$$

where t_{TCC} is the time-to-collision, T_{PC} is the pre-crash time constant that sets the minimum t_{TCC} where environmental observation is performed. Time constants T_1 and T_2 are used to shape the utility function for objective Ω'_3. An example utility function for Ω'_3 is given in Fig. 7.7.

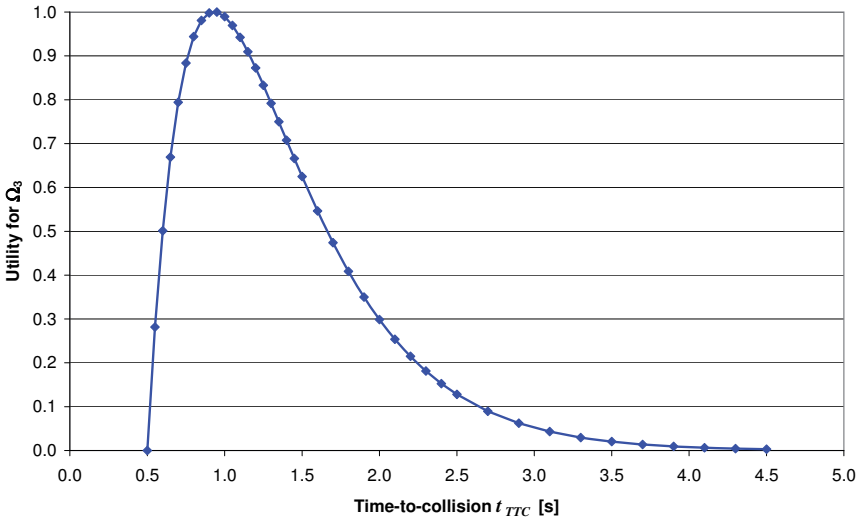


Figure 7.7: Normalised example utility function for objective Ω'_3. We use t_{PC} =0.5 s, T_1 =0.5 s, and T_2 =1.5 s.

The TTC of the different regions in our example road traffic scene is illustrated in Fig. 7.8.

Figure 7.8: TTC values for example road traffic scene. The minimum TTC inside each region is used as t_{TCC} in Tab. 7.13.

The minimum TTC inside each region is determined and used as t_{TCC} in Tab. 7.13.

A problem using TTC as a measure of situation criticality is that collision avoidance manoeuvres such as overtaking a slower car in front of the ego vehicle as illustrated in

Region	t_{TCC} [s]	Ω'_3
$\mathcal{R}_{1e}$	3.6	0.017
$\mathcal{R}_{23e}$	4.4	0.004
$\mathcal{R}_4$	1.1	0.942
$\mathcal{R}_{5e}$	1.1	0.942
$\mathcal{R}_6$	4.4	0.004
$\mathcal{R}_7$	3.6	0.017

Table 7.13: Example regions' time-to-collision values t_{TCC} and corresponding objective values Ω'_3 calculated using Eq. 7.9. As time constants t_{PC} =0.5 s, T_1 =0.5 s, and T_2 =1.5 s are assumed as in Fig. 7.7.

.

Fig. 7.9 are not considered. Models to evaluate situation criticality including avoidance manoeuvres are discussed in Kopischke *et al.* [199] and Mühlfeld *et al.* [200].

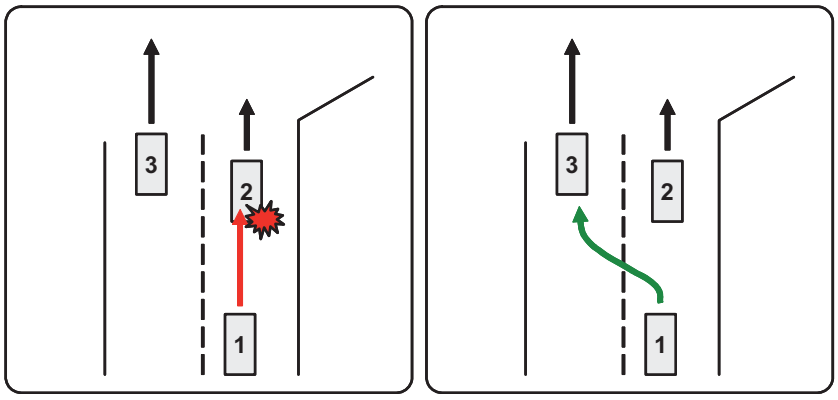


Figure 7.9: Collision avoidance manoeuvre changing a critical time-to-collision situation (left) into an uncritical overtaking manoeuvre (right). Source: Kopischke *et al.* [199].

Uncertainty Reduction

The objective of reducing uncertainty about a region in the environment relies on two assumptions. First, it is assumed that the system can be uncertain about the environment. Second, it is assumed that this uncertainty can be reduced.

During operation the system's uncertainty UC about the current environment is absolute at the beginning. By acquiring, processing, and interpreting exteroceptive sensor data, information can be gained. Following Shannon's definition

> *"... information is a measure of the decrement of uncertainty."* (Shannon [150])

Thus, semantic information about the environment in the form of a positive or negative detection or classification result reduces the uncertainty about the observed region. In our

proposed system an uncertainty range of $\mathcal{UC} = [0, 1]$ is used. An uncertainty value of $\mathcal{UC} = 0$ represents a state in which the system is absolutely certain that every traffic participant inside the observed region is detected and classified correctly. By contrast $\mathcal{UC} = 1$ represents a state in which the system is entirely uncertain about a region, which is the case for all regions prior to system operation.

It is difficult to determine the amount of information gained by acquiring, processing, and interpreting a region in the environment. As opposed to Ω_1 to Ω_3 which are determined from reasoning level data alone, uncertainty reduction further depends on sensor properties and classifier properties.

First, for sensor resource allocation the choice of sensor $\mathcal{S}$ has an influence on the quality of sensor data on which the classification is performed. In section 7.3.4 a sensor model is proposed to determine the utility $\mathcal{U}_{\mathcal{S}_i}(\mathcal{R}_m)$ of a sensor-region combination.

Second, for computational resource allocation the classifier's probability difference $\Delta P(\mathcal{TP}_n)$ between true positives $P(\mathcal{TP}_n|C)$ and false negatives $P(\mathcal{TP}_n|\neg C)$ is proposed as a measure for information in section 6.1.2. This information measure is again dependent upon the existence of classifiable traffic participants, which is taken into account in Eq. 7.10 using the traffic participant probability $P(\mathcal{TP}_n)$ as a factor.

$$\Delta P(\mathcal{TP}_n) = (P(\mathcal{TP}_n|C) - P(\mathcal{TP}_n|\neg C)) \cdot P(\mathcal{TP}_n) \tag{7.10}$$

Considering that the absolute reduction of uncertainty is both influenced and limited by the prior uncertainty, we propose using the uncertainty reduction measure in Eq. 7.11 as objective Ω'_4.

$$\Omega'_4 = \Delta\mathcal{UC} = \mathcal{UC} \cdot \sum_n \Delta P(\mathcal{TP}_n) \cdot \mathcal{U}_{\mathcal{S}_i}(\mathcal{R}_m) \tag{7.11}$$

If a region is not observed by our active vision system, the uncertainty about this region increases over time until it reaches $\mathcal{UC} = 1$. This increase over time is modelled in our proposed system applying a constant exponent $p = [0, 1]$ at every cycle k.

$$\mathcal{UC}_k = (\mathcal{UC}_{k-1})^p \tag{7.12}$$

This model of uncertainty increase is only an approximation. First, the increase of uncertainty depends upon the variability of the environment which cannot be expressed

with a fixed parameter p. Second, every region in the environment exhibits an individual variability which must be taken into account. It is also possible to use data level information such as motion vectors to estimate variability. In our system, the proposed exponential model is used due to its computational efficiency.

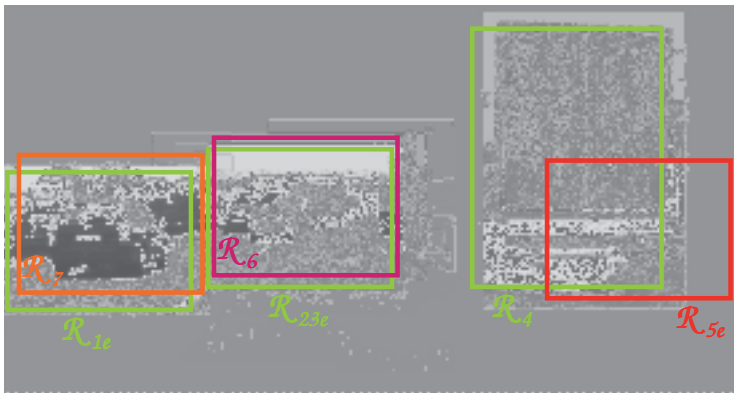


Figure 7.10: Current uncertainty about regions in the environment, reduced by previous observations and use of classifier cascades. $\mathcal{UC}$ is illustrated by reducing the contrast, resulting in a gray image for $\mathcal{UC} = 1$.

From the uncertainty map in Fig. 7.10 the mean uncertainty $\overline{\mathcal{UC}}$ for the example regions is obtained. Assuming an ideal sensor and an ideal classifier for our example, we use $\Delta P(\mathcal{IP}_n) = P_k^{\cup}(\mathcal{IP}_n)$, which is given in Tab. 7.6.

Region	$\overline{\mathcal{UC}}$	$\sum_n P(\mathcal{IP}_n) = P_k^{\cup}(\mathcal{IP}_n)$	Ω_4'
$\mathcal{R}_{1e}$	0.316	0.983	0.311
$\mathcal{R}_{23e}$	0.353	0.986	0.348
$\mathcal{R}_4$	0.692	0.983	0.680
$\mathcal{R}_{5e}$	0.781	0.937	0.732
$\mathcal{R}_6$	0.378	0.986	0.373
$\mathcal{R}_7$	0.368	0.983	0.362

Table 7.14: Example regions' mean uncertainty values $\overline{\mathcal{UC}}$ used to determine objective Ω_4'. For resource allocation, both sensor properties, computational resource properties must be considered (cf. Eq. 7.11).

Feasibility of Observation and Processing

In our active vision system, regions that can be observed, processed, and interpreted within a single cycle are preferable. As for uncertainty reduction, properties of both sensor resources and computational resources must be known to determine Ω_5.

For sensor resource allocation, an apparent example is $\mathcal{R}_4$, which exceeds the aperture angle of our high resolution sensors. In this case the utility of observing $\mathcal{R}_4$ with a high-resolution sensor is $\Omega_5 = 0$. Another example is a sensor requiring time for a gaze shift, such as a PTZ video camera. This is modelled by determining Ω'_5 to be

$$\Omega'_5 = \frac{1}{N_k} \tag{7.13}$$

where N_k is the number of system cycles k required to perform the gaze shift and acquire sensor level information about a region. In our presented system, all gaze shifts must be performed within a single cycle, effectively restricting the maximum gaze shift angle for PTZ sensors.

For computational resource allocation it must be ensured that classifier processes terminate within the given cycle time. The probability of termination within the maximum available time $P(t_n)$ is determined in Eq. 7.28 in section 7.4.2 on queue scheduling.

Objective Value Normalisation

Compiling the objective scores for the individual regions, a list of objectives Ω'_1 to Ω'_5 is shown in Tab.7.15.

Region	Ω'_1	Ω'_2	Ω'_3	Ω'_4	Ω'_5
$\mathcal{R}_{1e}$	0.742	46.9	0.017	0.311	1.00
$\mathcal{R}_{23e}$	0.745	36.8	0.004	0.348	1.00
$\mathcal{R}_4$	0.742	44.1	0.942	0.680	0.00
$\mathcal{R}_{5e}$	0.747	42.4	0.942	0.732	1.00
$\mathcal{R}_6$	0.745	38.6	0.004	0.373	1.00
$\mathcal{R}_7$	0.742	53.3	0.017	0.362	1.00

Table 7.15: Objectives' utility values for example regions.

In order to compare objectives across dimensions, a normalisation of objective values is performed. For this, the maximum utility an objective can assign is normalised to 1.00. This ensures an equal weight to all objectives.

$$\Omega_m(\mathcal{R}_n) = \frac{\Omega'_m(\mathcal{R}_n)}{\max\left(\Omega'_m(\mathcal{R}_n)\right)} \tag{7.14}$$

For our example candidate regions from Tab. 7.15 the normalised objective values using Eq. 7.14 are given in Tab. 7.16.

Region	Ω_1	Ω_2	Ω_3	Ω_4	Ω_5
$\mathcal{R}_{1e}$	0.994	0.880	0.018	0.424	1.000
$\mathcal{R}_{23e}$	0.997	0.690	0.004	0.476	1.000
$\mathcal{R}_4$	0.994	0.827	1.000	0.930	0.000
$\mathcal{R}_{5e}$	1.000	0.795	1.000	1.000	1.000
$\mathcal{R}_6$	0.997	0.724	0.004	0.509	1.000
$\mathcal{R}_7$	0.994	1.000	0.018	0.494	1.000

Table 7.16: Normalised objectives' utility values for example regions from Tab. 7.15.

Determination of Combined Utility

The normalised objectives' utility values in Tab. 7.16 are used to determine the combined utility $\mathcal{U}(\mathcal{R}_m)$ for every region. For this, the different utility concepts discussed in section 2.3.3 are used. The combined utility values in Tab. 7.17 show that region $\mathcal{R}_{5e}$ is estimated to be the optimum region for a focused sensor by all utility concepts. This is comprehensible as a detected human traffic participant inside a salient region with a short time-to-collision and a high current uncertainty must be considered highly critical.

Region	$\mathcal{U}^u$	$\mathcal{U}^\times$	$\mathcal{U}^e$	$\mathcal{U}^\$$	$\mathcal{U}^\lambda$
$\mathcal{R}_{1e}$	3.316	0.0067	0.018	<u>1.000</u>	0.018,0.457,..
$\mathcal{R}_{23e}$	3.168	0.0014	0.004	<u>1.000</u>	0.004,0.690,..
$\mathcal{R}_4$	3.750	0.0000	0.000	<u>1.000</u>	0.000,0.570,..
$\mathcal{R}_{5e}$	<u>4.795</u>	<u>0.7955</u>	<u>0.795</u>	<u>1.000</u>	<u>0.795,0.941,..</u>
$\mathcal{R}_6$	3.235	0.0016	0.004	<u>1.000</u>	0.004,0.724,..
$\mathcal{R}_7$	3.506	0.0089	0.018	<u>1.000</u>	0.018,0.531,..

Table 7.17: Combined utility using different utility concepts for the normalised objectives given in Tab. 7.16. Optimum combined utility values $\mathcal{U}(\mathcal{R}_m)^*$ are underlined.

7.2.3 Evaluation of Combined Utility

As all multi-objective utility concepts combine the objectives differently, a common score must be determined for evaluation. This score determines the criticality of a candidate region using the observed region's TTC score, its FMEA score, and its present uncertainty.

The main difference between the criticality score and the combined utility concepts is, that the criticality score is calculated using ground truth information as opposed to the objectives using semantic information determined by the active vision system. This also allows the use of a utility concept as a score calculation scheme without biasing the results in favour of the chosen utility concept.

The criticality score $\eta(\mathcal{R}_m)$ is determined using the Nash product of the FMEA score $\Omega_{1_{GT}}$ using ground truth information, the candidate region's TTC score Ω_3, and the present uncertainty Ω_4 about the candidate region $\mathcal{R}_m$.

$$\eta(\mathcal{R}_m) = \Omega_{1_{GT}}(\mathcal{R}_m) \cdot \Omega_3(\mathcal{R}_m) \cdot \Omega_4(\mathcal{R}_m) \tag{7.15}$$

The Nash product is chosen due to its preference of a balanced set of high individual abilities (see section 2.3.3). Erroneous objective values close to zero, which present a problem for Nash product calculation, are less frequent due to the use of ground truth information as opposed to measured data. Three sequences (TRC, URB, and MWY, cf. appendix A) are used to evaluate the different combined utility concepts with our criticality score η. The results of selecting a single region are given in Fig. 7.11.

From Fig. 7.11 it can be seen that all combined utility concepts besides Elitist utility select candidate regions with a criticality score η in the range of two to four times larger as compared to a random candidate selection. The Elitist utility concept fails to balance the different objectives against each other resulting in a smaller criticality score η. As expected, the mean TTC scores and FMEA scores are smaller if the present uncertainty UC is considered (hatched bars) due to the additional objective in the balancing.

From the results in Fig. 7.11 it can be inferred that for the selection of a single region, Egalitarian utility shows marginally better results as compared to Utilitarian utility, Nash product utility, and Leximin utility. These findings are also shown in earlier work using a different criterion for evaluation in Matzka *et al.* [12]. However, an evaluation of the complete contextual resource allocation system with multiple sensor resources in section 7.5.1 results in a different ranking (cf. Tab. 7.42). There, the use of uncertainty information for resource allocation also shows a significant increase of the mean criticality score η.

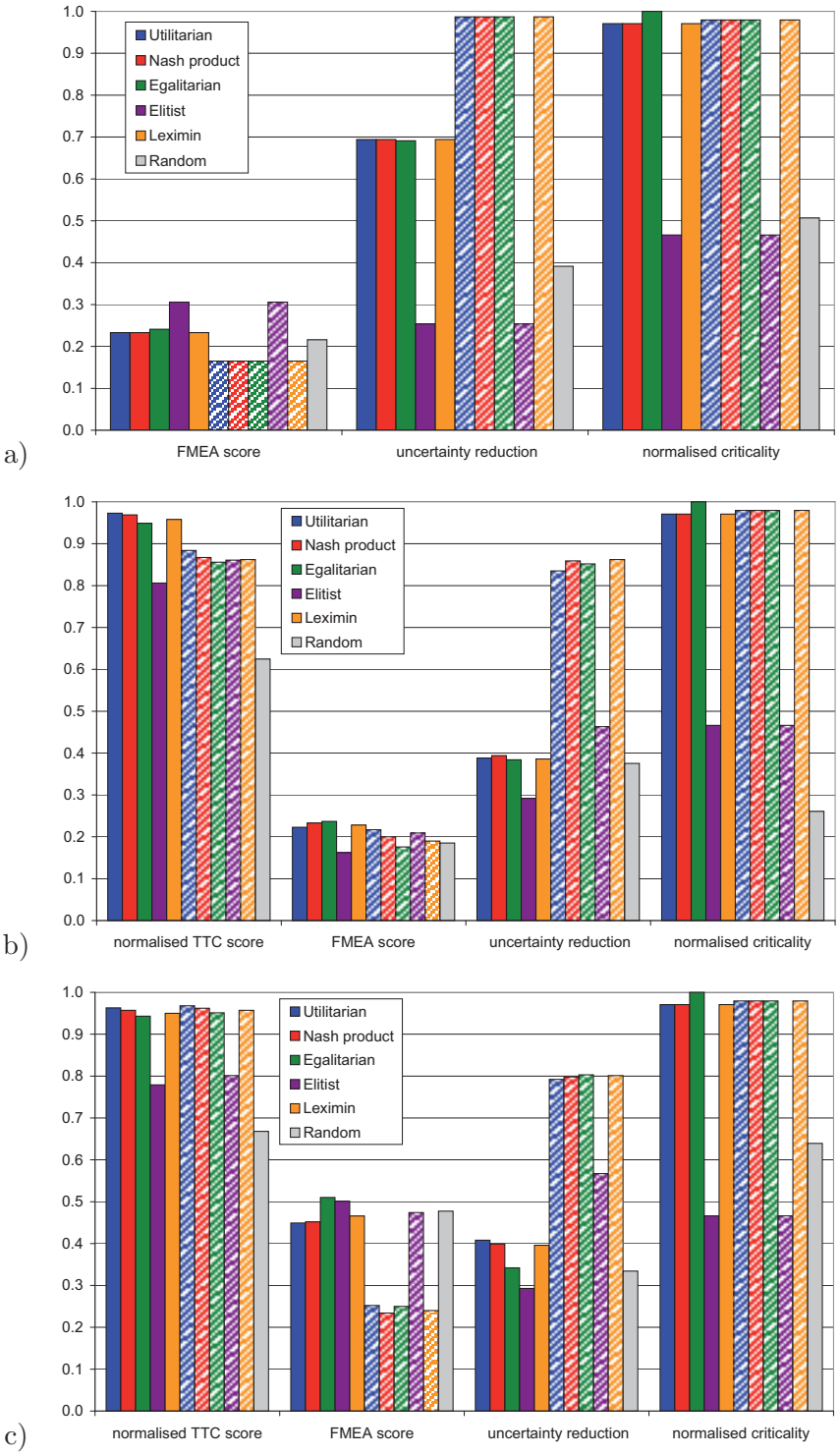


Figure 7.11: Resulting mean TTC score (where available), mean FMEA score, mean uncertainty reduction, and mean overall criticality for three sequences using different combined utility concepts. Solid bars indicate that no uncertainty information is considered in the combined utility, which is the case for hatched bars. Gray bars indicate a random selection among the candidate regions. Figure a) shows the results for the traffic calmed road sequence (TRC), b) for the urban road sequence (URB), and c) for the motorway sequence (MWY).

7.3 Sensor Resource Allocation Heuristics

The combined utility to observe each individual region is determined using the methods described in section 7.2. In a next step, sensor resources are allocated individual regions. Regions and sensors can be thought of as in a 1-to-n relationship, i.e. each region can be observed by multiple sensors, but each sensor can only observe a single region. Alternatively it is possible to restrict the allocation of sensors to a maximum of one sensor per region, which then constitutes a 1-to-1 relationship. In the latter case with a maximum of one sensor per region, for $N_{\mathcal{S}}$ sensors and $N_{\mathcal{R}}$ regions there exist

$$N_{\mathcal{A}^1} = \frac{N_{\mathcal{R}}!}{(N_{\mathcal{R}} - N_{\mathcal{S}})!} \tag{7.16}$$

possible allocations. This factorial growth in complexity is demonstrated for different combinations of $N_{\mathcal{S}}$ and $N_{\mathcal{R}}$ in Tab. 7.18. For a 1-to-n relationship multiple sensors are allowed to be assigned a single region. This results in a total of

$$N_{\mathcal{A}} = (N_{\mathcal{R}})^{N_{\mathcal{S}}} \tag{7.17}$$

possible allocations. This exponential growth in complexity is also demonstrated for different combinations of $N_{\mathcal{S}}$ and $N_{\mathcal{R}}$ in Tab. 7.18.

	$N_{\mathcal{S}} = 2$	$N_{\mathcal{S}} = 3$	$N_{\mathcal{S}} = 4$
$N_{\mathcal{R}} = 5$	25 (20)	125 (60)	625 (120)
$N_{\mathcal{R}} = 10$	100 (90)	1,000 (720)	10,000 (5,040)
$N_{\mathcal{R}} = 15$	225 (210)	3,375 (2,730)	50,625 (32,760)
$N_{\mathcal{R}} = 20$	400 (380)	8,000 (6,840)	160,000 (116,280)

Table 7.18: Possible allocations for multiple sensors per resource $N_{\mathcal{A}}$ and single sensors per resource in parentheses ($N_{\mathcal{A}^1}$).

The computational cost to find the optimal allocation $\mathcal{A}^*$ is therefore dependent upon the number of regions and sensors as well as the number of sensors allowed to be allocated on a single region. However, our proposed system has limited resources and has to satisfy real-time constraints.

In the following, different search methods to find the allocation with optimum overall combined utility are proposed. For this, we assume a basic system with $N_{\mathcal{S}} = 3$ real sensors and $N_{\mathcal{R}} = 3$ regions. The example utility values for every sensor-region combination are given in Tab. 7.19.

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_{12}$	$\mathcal{S}_{13}$	$\mathcal{S}_{23}$	$\mathcal{S}_{123}$
$\mathcal{R}_1$	9	10	9	12	11	12	13
$\mathcal{R}_2$	1	9	3	9	4	10	11
$\mathcal{R}_3$	5	8	4	9	9	9	10

Table 7.19: Example overall combined utility values for different sensor-region combinations assuming $N_{\mathcal{S}} = 3$ real sensors and $N_{\mathcal{R}} = 3$.

7.3.1 Exhaustive Search Method

It is possible to determine the overall combined utility exhaustively for every feasible allocation. If the determination is completed within a predefined maximum search time $t_{s_{max}}$, the allocation with maximum overall combined utility is known. It can be seen from Tab. 7.18 that this is only possible for small values of $N_{\mathcal{S}}$ and $N_{\mathcal{R}}$. If the time required for the exhaustive search exceeds $t_{s_{max}}$, the algorithm is terminated preemptively and the best overall combined utility found before termination is considered to be $\mathcal{A}^*$.

The probability to find the global optimum $\mathcal{A}^*$ is less than 1.00 if only a subset of all possible allocations is considered. Assuming an equal overall combined utility distribution over all resource-sensor combinations, the probability to find the global optimum $\mathcal{A}^*$ within $t_{s_{max}}$ is

$$P(\mathcal{A}^*|t_{s_{max}}) = \frac{N_{\mathcal{A}}(t_{s_{max}})}{N_{\mathcal{A}}} \tag{7.18}$$

Using the example utility values from Tab. 7.19, the determination of $\mathcal{A}^*$ using exhaustive search is shown in Tab. 7.20. There, the optimum allocation with $\mathcal{U} = 23$ is found as late as step 22 from a total of 27 search steps.

Step	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{U}$
1	$\mathcal{R}_1$	$\mathcal{R}_1$	$\mathcal{R}_1$	13*
2	$\mathcal{R}_1$	$\mathcal{R}_1$	$\mathcal{R}_2$	15*
3	$\mathcal{R}_1$	$\mathcal{R}_1$	$\mathcal{R}_3$	16*
4	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_1$	20*
5	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_2$	19
⋮		⋮		⋮
22	$\mathcal{R}_3$	$\mathcal{R}_2$	$\mathcal{R}_1$	23*
⋮		⋮		⋮
27	$\mathcal{R}_3$	$\mathcal{R}_3$	$\mathcal{R}_3$	10

Table 7.20: Exhaustive search for the sensor-region combination allocation with maximum overall combined utility. The global maximum of 23 is found after 22 calculations and six local maxima (indicated with an *).

An option to increase the probability of finding the global optimum allocation $\mathcal{A}^*$ within $t_{s_{max}}$ is to use a graph search strategy as proposed by Krahnstoever *et al.* [201] as described below. Alternatively the use of dynamic programming, i.e. the simplification of a complex problem by recursively breaking it down into subproblems of lower complexity, can be considered.

7.3.2 Best-First Search Method

Using a best-first search approach for a closely related problem is proposed by Krahnstoever *et al.* [201]. For the example utility values given in Tab. 7.19 the best-first search strategy illustrated in Fig. 7.12 yields an overall combined utility of 19 after the first search step.

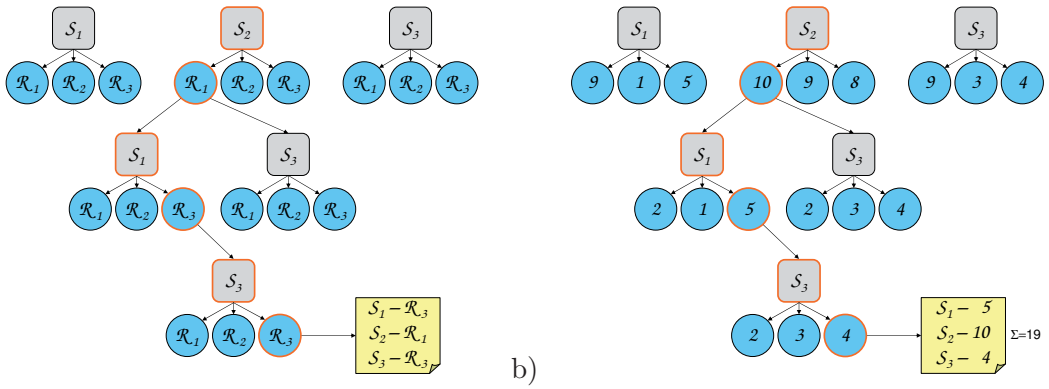


Figure 7.12: Example best-first search path for three sensor resources $\mathcal{S}_{1..3}$ and candidate regions $\mathcal{R}_{1..3}$ (a) with its individual utility gains given in (b).

The best-first search algorithm is a depth-first search method. The search path is determined by selecting the local optimum choice at every node. If a complete solution is found, best-first search traces back its path and searches for alternative superior allocations either until all possible allocations are evaluated or until $t_{s_{max}}$ is reached. Best-first graph search is a greedy algorithm and thus prone to failing to find the global optimum due to early decisions that lead towards a local instead of a global optimum. Moreover a graph search has to ensure that no sensor-region combination is evaluated twice.

7.3.3 Pre-Sorted Search Method

Our evaluation of search heuristics in section 7.3.5 shows that the global optimum is not always found within a short search time $t_{s_{max}}$ by either exhaustive search or best-first

search. We propose a pre-sorted search method that increases the probability of finding the global optimum within a certain timespan by performing the following steps:

1. Calculate the combined utilities for all region-sensor combinations.
2. Sort the preferred regions for each sensor in order of descending utility.
3. Calculate the overall combined utility using every sensors' most preferred region.
4. Consecutively degrade the preference ranks for one or multiple sensors until reaching the least preferred region for every sensor in a breadth-first manner.

The search strategy is to incrementally increase the sum of individual rank degradations and to update the allocation currently presumed optimal by any superior subsequent allocation. The ranks for our example utility values in Tab. 7.19 are given in Tab. 7.21.

Rank	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$
0	$\mathcal{R}_1$	$\mathcal{R}_1$	$\mathcal{R}_1$
1	$\mathcal{R}_3$	$\mathcal{R}_2$	$\mathcal{R}_3$
2	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_2$

Table 7.21: Rank table for example utility values given in Tab. 7.19.

For three sensors $\mathcal{S}_{1..3}$ from Fig. 7.12 the rank degradation table is given in Tab. 7.22 alongside the resulting sensor-regions combinations and the resulting overall combined utility. In this thesis, a rank degradation table is defined to be a list of ranks increasing from the most preferred rank (i.e. 0) for all sensors towards the least preferred rank for all sensors.

	Rank			Allocation			
$\sum$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{U}$
0	0	0	0	$\mathcal{R}_1$	$\mathcal{R}_1$	$\mathcal{R}_1$	13*
1	1	0	0	$\mathcal{R}_3$	$\mathcal{R}_1$	$\mathcal{R}_1$	17*
1	0	1	0	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_1$	21*
1	0	0	1	$\mathcal{R}_1$	$\mathcal{R}_1$	$\mathcal{R}_3$	16
2	2	0	0	$\mathcal{R}_2$	$\mathcal{R}_1$	$\mathcal{R}_1$	13
2	1	1	0	$\mathcal{R}_3$	$\mathcal{R}_2$	$\mathcal{R}_1$	23*
2	1	0	1	$\mathcal{R}_3$	$\mathcal{R}_1$	$\mathcal{R}_3$	15
2	0	2	0	$\mathcal{R}_1$	$\mathcal{R}_3$	$\mathcal{R}_1$	19
2	0	1	1	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	22
⋮		⋮			⋮		⋮
6	2	2	2	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_2$	12

Table 7.22: Rank degradation table for $\mathcal{S}_{1..3}$ with resulting sensor-region combinations and resulting overall combined utility $\mathcal{U}$. The global maximum of 23 is found after six calculations and three local maxima (indicated with an *).

Generation of a Rank Degradation Table

The generation of a rank degradation table is a problem of partitioning a given integer number, the sum of rank degradations, into a set of slots, the sensors. Further all distributions have to be distinguishable. Both problems of partitioning integer numbers and distinguishably permuting the resulting sets are classic problems[1].

An algorithm to solve the problem of partitioning n indistinguishable objects into k distinguishable slots of complexity

$$O(n,k) = \binom{n+k-1}{n}$$

is presented by Fenichel [203] and corrected by Gray [204]. We adapt this algorithm to generate the desired rank degradation table.

Input: A vector *bin* of length $N_{\mathcal{S}}$, the number of sensors $N_{\mathcal{S}}$,
and the number of regions $N_{\mathcal{R}}$
Output: A list of *bin* vectors
for *(*$i \leftarrow 0$ **to** ($N_{\mathcal{S}}$ * ($N_{\mathcal{R}}$-1))* **do**
  $bin(0) \leftarrow i$;
  $bin(1, .., N_{\mathcal{S}}) \leftarrow 0$;
  if *(*$i < N_{\mathcal{S}}$*)* **then**
    append *bin* to solutions
  end
  call partition(*bin*, 0, $N_{\mathcal{R}} - 1$)
end

Algorithm 7.1: Iterative calls to the recursive 'partition' algorithm Alg. 7.2.

The algorithm used for the generation of the degradation table is recursive. It continually increments the current slot's right-hand neighbour at the actual slot's expense. This is performed until either the actual slot is empty, the right-hand neighbour has reached the maximum degradation level or the last slot is reached. The implemented partitioning algorithm can be seen in Alg. 7.1 and 7.2. A partial example result for this partitioning algorithm is shown in Tab. 7.22, where three regions are to be allocated to three sensors.

[1] In 1669, the problem of partitioning a number is raised by Gottfried Wilhelm Leibniz in a letter to Johann Bernoulli, remarking that the problem seemed difficult yet important (cf. Dickson [202]).

```
Input: A vector bin of length N_S, a position pos, and a maximum value max
Output: A list of bin vectors
if ((pos + 1) < N_S) then
    while (bin(pos) > 0) do
        decrease bin(pos) by 1;
        increase bin(pos + 1) by 1;
        if (bin(pos) ≤ max) then
            if (bin(pos + 1) ≤ max) then
                append bin to solutions;
            end
            call partition(bin, (pos + 1), max);
        end
    end
end
```

Algorithm 7.2: Recursive 'partition' algorithm.

7.3.4 Sensor Model for Utility Calculation

Every sensor in the system exhibits individual properties, extending the other sensors' abilities. Mandatory sensors properties in our system are the sensor's observable field of view and the classification rates for individual traffic participants $P(C|\mathcal{TP}_n)$ and $P(C|\neg\mathcal{TP}_n)$.

The above properties are dependent upon a range of sensor characteristics which are represented in the system as optional properties such as resolution, modalities, pan angle, tilt angle, and zoom levels as applicable.

The utility of any sensor-region combination can be seen as the reduction of uncertainty about traffic participants inside an observed region. For every region, a fused traffic participant probability $P_k^{\cup}(\mathcal{TP}_n)$ exists. With this and with knowledge about the maximum attainable detection and classification performance $P(C|\mathcal{S}_\iota, \mathcal{TP}_n)$ for a sensor $\mathcal{S}_\iota$, the utility $\mathcal{U}(\mathcal{R}_m, \mathcal{S}_\iota)$ of observing region $\mathcal{R}_m$ with sensor $\mathcal{S}_\iota$ is determined by using Eq. 7.19 and 7.20 to recalculate $\Omega_4'(\mathcal{R}_m, \mathcal{S}_\iota)$.

$$\Omega_4'(\mathcal{R}_m, \mathcal{S}_\iota) = \Omega_4'(\mathcal{R}_m) \cdot \mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m) \tag{7.19}$$

with

$$\mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m) = \sum_n \left(P(C|\mathcal{S}_\iota, \mathcal{TP}_n) \cdot P_k^{\cup}(\mathcal{TP}_n) \right) \tag{7.20}$$

Utility Calculation for $N_{\mathcal{A}^1}$

For the case that every region is assigned a maximum of one sensor, it is useful to generate an $(N_{\mathcal{S}} \times N_{\mathcal{R}})$ lookup table, that contains each region's combined utility $\mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m)$ if it is observed by sensor $\mathcal{S}_\iota$ (e.g. Tab. 7.19). As different sensors observing different regions constitute independent actions, the overall sensor utility $\mathcal{U}_{\mathcal{A}^1_{\mathcal{S}}}$ is the sum of the sensors' individual combined utilities.

$$\mathcal{U}_{\mathcal{A}^1_{\mathcal{S}}} = \sum_{\iota=1,\ldots,N_{\mathcal{S}}} \mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m) \tag{7.21}$$

Utility Calculation for $N_{\mathcal{A}}$

If multiple sensors are allowed to be assigned a single region, the set of overall combined utilities for a single sensor per region $\mathcal{U}_{\mathcal{A}^1_{\mathcal{S}}}$ has to be extended for the cases where multiple sensors are assigned at least one region. For this case, the notion of *mutual information* is relevant.

Mutual information is a measure for the mutual dependence of n random variables, calculated from the joint probability distribution of m sets of data. Mutual information is proposed for registration of multiple images by Viola and Wells [205, 206], and Collignon *et al.* [207] independently. For a set of aligned images, the joint probability distribution, and thus the mutual information, is maximised. A dispersed joint probability distribution is indicative of little mutual information and thus a poorly registered set of images.

While the presence of mutual information is beneficial for image registration of multiple camera sources, it also diminishes the increase of information if multiple similar sensors are observing the same region from the same viewpoint. If a multi-modal sensor combination is allocated to a single region, e.g. a light intensity sensor and a range sensor, the amount of mutual information may be little and the observation of the same region by multiple sensors therefore be desirable (cf. Gould *et al.* [208]). In Fig. 7.13 common cases for mutual information (e.g. contour) and non-mutual information (e.g. texture) for luminance-range combinations are demonstrated.

The use of two calibrated and registered video-cameras with a wide baseline is an example for an emergent modality, i.e. a range image, that can be gained by observing the same region with two cameras at the same time instead of focusing different regions with two sensors. The presented case of binocular stereo vision is a known application in

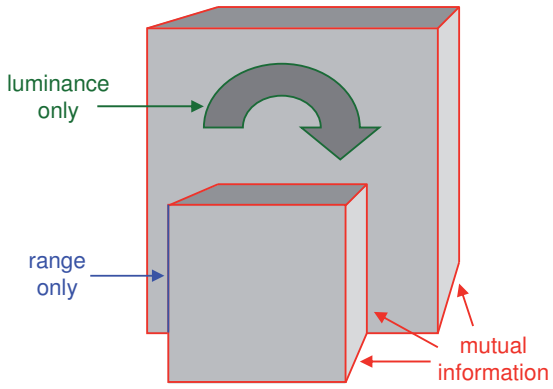


Figure 7.13: Example scene with two adjoined boxes. A high degree of mutual information (red) can be observed. However, the flat arrow can only be detected using luminance information (green) and part of the left corner of the box in front can only be detected using range information (blue).

computer vision and is described in early works such as Lucas and Kanade [209]. Returning to the problem of increasing utility gained by observing a region, it is necessary to observe the influence of different sensor modalities to the attainable utility, which is illustrated in Fig. 7.14.

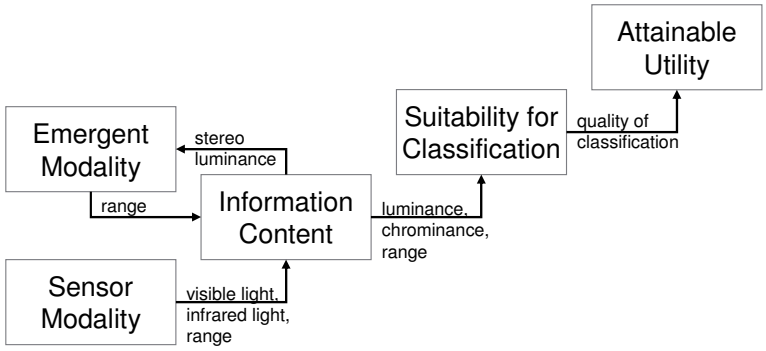


Figure 7.14: Influence of different sensor modalities on the level of attainable utility, including the effect of emergent modality for the case of two calibrated and registered video cameras with a wide baseline.

The calculation of the information gain by adding a second sensor to the first sensor is dependent on the sensor configuration and can be grouped into a set of three classes:

- additional sensors providing mutual information
- additional sensors providing additional information
- additional sensors providing emergent information

The relation between information and modalities is discussed below.

Identical Modalities

If multiple sensors observe the same modality, e.g. visual light intensity, a high degree of mutual information must be assumed. In this case the sensor with the highest utility for a candidate region $\mathcal{U}^*_{\mathcal{S}}$ also defines the overall utility.

$$\mathcal{U}_{\mathcal{A}_{\mathcal{S}}} = \max_{\iota=1,\ldots,N_{\mathcal{S}}} (\mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m)) \tag{7.22}$$

Differing Modalities

For multiple sensors observing different modalities the degree of mutual information depends both on the sensors and the observed objects (cf. Fig. 7.13, Gould *et al.* [208]). Assuming mutual independence of all sensors, the knowledge gain by observation can be calculated as

$$\mathcal{U}_{\mathcal{A}_{\mathcal{S}}} = 1 - \prod_{\iota=1..N_{\mathcal{S}}} (1 - \mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m)) \tag{7.23}$$

Additional Modalities and Emergent Modalities

Multiple sensors sharing at least one modality but differing in others are an example for additional modalities. An example for this is the combination of a grayscale camera and a colour camera. If the grayscale camera exhibits a higher utility, the colour channel of the colour camera can provide additional information which can provide additional utility. However, if the colour camera exhibits a higher utility, the grayscale camera does not provide an additional modality and therefore does not increase the allocation's overall utility.

Emergent modalities constitute a special case of additional modalities, where the resulting set of modalities exceeds the joint set of modalities for all sensors. A common example for an emergent modality is a range image, that is acquired by observing the same region with two video cameras at the same time. In these cases, range information emerges by combining two sets of luminance data.

Calculating the utility gain of the n^{th} sensor is difficult for additional and emergent modalities. It can be attempted by introduction a mutual information factor into Eq. 7.23 to account for mutual information that does not increase the utility of observation. However, in this thesis a virtual sensor concept is proposed.

Virtual Sensor Concept

Due to the difficulty of calculating the utility of using a combination of sensors, we resort to a concept using virtual sensors. A virtual sensor is a combination of existing sensors that integrates information about all combined attributes such as modalities or resolution.

Virtual sensors are indicated by multi-digit indices. The virtual sensor $\mathcal{S}_{12}$ therefore denotes the combination of sensor $\mathcal{S}_1$ and $\mathcal{S}_2$. For three sensors $\mathcal{S}_1$ to $\mathcal{S}_3$ there exist four virtual sensors $\mathcal{S}_{12}$, $\mathcal{S}_{23}$, $\mathcal{S}_{13}$, and $\mathcal{S}_{123}$ given that all sensors have at least one region that can be observed at the same time. Virtual sensors also have a field of view, which is the overlapping field of view of all combined existing sensors. Example utility value tables for three existing and four virtual sensors are given in Tab. 7.19 and Tab. 7.23.

The virtual sensor concept is preferable to calculating joint utilities during runtime in three respects. First, it is easy to implement, as an allocation of different existing sensors towards the same region is mapped onto an allocation of one virtual sensor towards this region. Second, using a virtual sensor allows fusion of data at all data levels from sensor data level to syntactical level without further consideration during region-sensor allocation. Third, virtual sensors reduce computational costs during runtime, as only a single sensor's utility has to be computed as opposed to computing and combining multiple utility values.

The disadvantage of a virtual sensor concept is the necessity to extend the set of virtual sensors every time a new sensor is introduced into the sensor system. If all sensors have at least one observable region in common, the number of virtual sensors to be introduced equals the Eulerian number for $N_{\mathcal{S}}$ and can be calculated using Eq. 7.24 [210].

$$N_{\mathcal{S}_{virtual}} = \left\langle \begin{array}{c} N_{\mathcal{S}} \\ 1 \end{array} \right\rangle = 2^{N_{\mathcal{S}}} - N_{\mathcal{S}} - 1 \tag{7.24}$$

The Eulerian number sequence for $N_{\mathcal{S}}$ =1,..,5 is 0, 1, 4, 11, 26. For $N_{\mathcal{S}} > 5$ the number of virtual sensors quickly becomes very large. In practice however this is seldom a problem, as the number of sensors with a common field of view rarely exceeds four different sensors in automotive applications.

7.3.5 Evaluation of Sensor Resource Allocation Heuristics

In sections 7.3.1 to 7.3.3 three algorithms to allocate candidate regions to sensor resources are described: exhaustive search, best-first search, and pre-sorted search.

Test Conditions

The evaluation is conducted using data from $2{\cdot}10^3$ test runs for each individual $N_{\mathcal{S}} = [2, 5]$, $N_{\mathcal{R}} = [2, 20]$ combination. Utility values for sensor-region combinations are randomly generated using an equal distribution in [0,1]. Assuming independence of all sensors, the utility of a virtual sensor $\mathcal{U}(\mathcal{S}_{\vec{n}}, \mathcal{R}_m)$ with $\vec{n} \subseteq 1, .., N_{\mathcal{S}}$ for a single region $\mathcal{R}_m$ is calculated using Eq. 7.25.

$$\mathcal{U}_{\mathcal{S}_{\vec{n}}}(\mathcal{R}_m) = 1 - \prod_{\iota \in \vec{n}} (1 - \mathcal{U}_{\mathcal{S}_\iota}(\mathcal{R}_m)) \tag{7.25}$$

In Tab. 7.23 an example utility lookup table is generated using random values for sensors $\mathcal{S}_1$ to $\mathcal{S}_3$ and utility values for virtual sensors $\mathcal{S}_{12}$ to $\mathcal{S}_{123}$ calculated using Eq. 7.25.

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_{12}$	$\mathcal{S}_{23}$	$\mathcal{S}_{13}$	$\mathcal{S}_{123}$
$\mathcal{R}_1$	0.18	0.54	0.91	0.62	0.96	0.93	0.97
$\mathcal{R}_2$	0.09	0.35	0.59	0.42	0.74	0.63	0.76
$\mathcal{R}_3$	0.37	0.70	0.24	0.81	0.78	0.53	0.86
$\mathcal{R}_4$	0.71	0.59	0.92	0.88	0.97	0.98	0.99
$\mathcal{R}_5$	0.17	0.61	0.22	0.68	0.70	0.35	0.75

Table 7.23: Example random utility value table for sensors $\mathcal{S}_1$ to $\mathcal{S}_3$ and utility values for virtual sensors $\mathcal{S}_{12}$ to $\mathcal{S}_{123}$ calculated using Eq. 7.25.

Exhaustive Search

Exhaustive search is evaluated using 2,000 test runs as defined in section 7.3.5. As expected, the mean number and maximum number of search steps increase approximately proportional to the total number of feasible allocations $N_{\mathcal{A}} = N_{\mathcal{R}}^{N_{\mathcal{S}}}$.

	$N_{\mathcal{R}} = 2$	$N_{\mathcal{R}} = 5$	$N_{\mathcal{R}} = 10$	$N_{\mathcal{R}} = 15$	$N_{\mathcal{R}} = 20$
Minimum	1	12	12	14	23
Mean ($0.5 \cdot N_{\mathcal{A}}$)	4 (4)	62 (63)	488 (500)	1,628 (1,688)	3,700 (4,000)
Maximum ($N_{\mathcal{A}}$)	8 (8)	114 (125)	989 (1,000)	3,363 (3,375)	7,982 (8,000)

Table 7.24: Minimum, mean, and maximum search steps at which the global maximum is found for the allocation of $N_{\mathcal{S}}$=3 sensor resources and $N_{\mathcal{R}}$ regions using exhaustive search. Both the mean number and maximum number of search steps increase approximately proportional to the total number of feasible allocations $N_{\mathcal{A}}$ (given in parentheses).

Besides the number of search steps required to find the global maximum, the convergence of the best local maximum utility towards the global maximum utility is of interest.

This convergence is given in Fig. 7.15 as the minimum, mean, and maximum fraction of the best known solution of the global maximum.

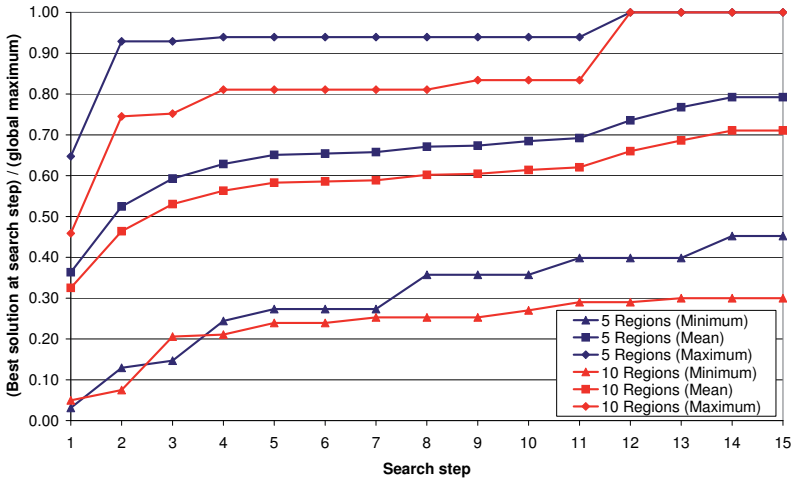


Figure 7.15: Minimum △, mean □, and maximum ♢ fraction of the best known solution of the global maximum at a certain search step using exhaustive search for $N_{\mathcal{S}}$=3 sensor resources.

Best-First Search

The best-first search method proposed in section 7.3.2 is evaluated under the test conditions defined in section 7.3.5. It can be seen from Fig. 7.16 that the probability of finding the global maximum is generally good for large number of regions. However best-first search fails to converge to the global maximum of 1.00 within the first eight search steps, which can also be seen in Tab. 7.25.

The minimum, mean, and maximum numbers of search steps at which the global maximum is found using the best-first algorithm are shown in Tab. 7.25. The mean number of search steps is shown to scale largely with N_R.

	$N_{\mathcal{R}} = 2$	$N_{\mathcal{R}} = 5$	$N_{\mathcal{R}} = 10$	$N_{\mathcal{R}} = 15$	$N_{\mathcal{R}} = 20$
Minimum	1	1	1	1	1
Mean	2.5	8.0	11.9	18.0	20.7
Maximum	8	56	202	466	802

Table 7.25: Minimum, mean, and maximum number of search steps at which the global maximum is found for the allocation of $N_{\mathcal{S}}$=3 sensor resources and $N_{\mathcal{R}}$ regions using best-first search.

The minimum and mean fraction of the best known solution of the global maximum are drawn for the first 15 search steps can be seen in Fig. 7.17.

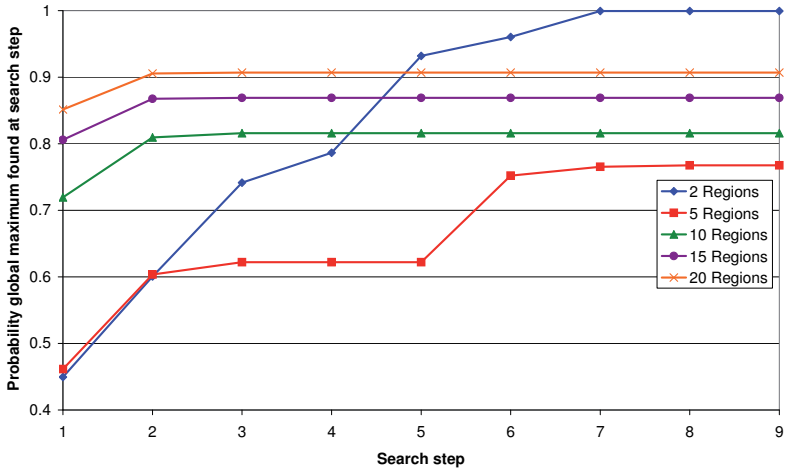


Figure 7.16: Probability of best-first search algorithm to find the global maximum at a certain step using $N_{\mathcal{S}}$=3 sensor resources.

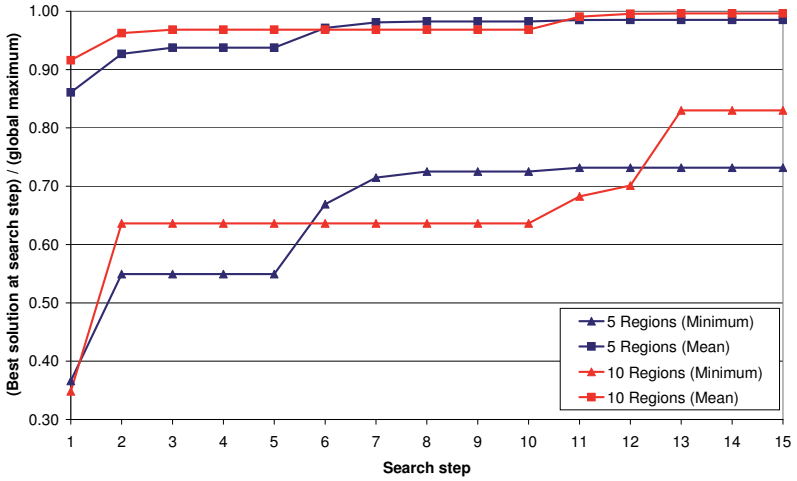


Figure 7.17: Minimum △ and mean □ fraction of the best known solution of the global maximum at a certain search step using best-first search for $N_{\mathcal{S}}$=3 sensor resources.

Pre-sorted Search

The pre-sorted search method proposed in section 7.3.3 is evaluated under the test conditions defined in section 7.3.5. It can be seen from Fig. 7.18 that the probability of finding the global maximum rapidly converges to 1.00 using only a small number of search steps.

For $N_{\mathcal{S}}$=3 sensor resources the minimum, mean, and maximum search steps at which the global maximum is found can be seen from Tab. 7.26.

The minimum and mean fraction of the best known solution of the global maximum are drawn for the first 15 search steps can be seen in Fig. 7.19.

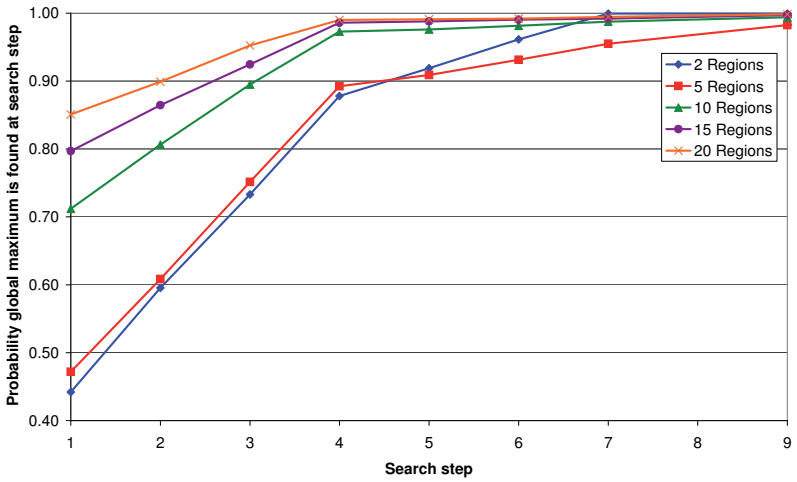


Figure 7.18: Probability of pre-sorted search algorithm to find the global maximum at a certain step using $N_{\mathcal{S}}$=3 sensor resources.

	$N_{\mathcal{R}} = 2$	$N_{\mathcal{R}} = 5$	$N_{\mathcal{R}} = 10$	$N_{\mathcal{R}} = 15$	$N_{\mathcal{R}} = 20$
Minimum	1	1	1	1	1
Mean	2.5	2.6	1.7	1.5	1.3
Maximum	7	19	19	10	10

Table 7.26: Minimum, mean, and maximum search steps at which the global maximum is found for the allocation of $N_{\mathcal{S}}$=3 sensor resources and $N_{\mathcal{R}}$ regions using pre-sorted search.

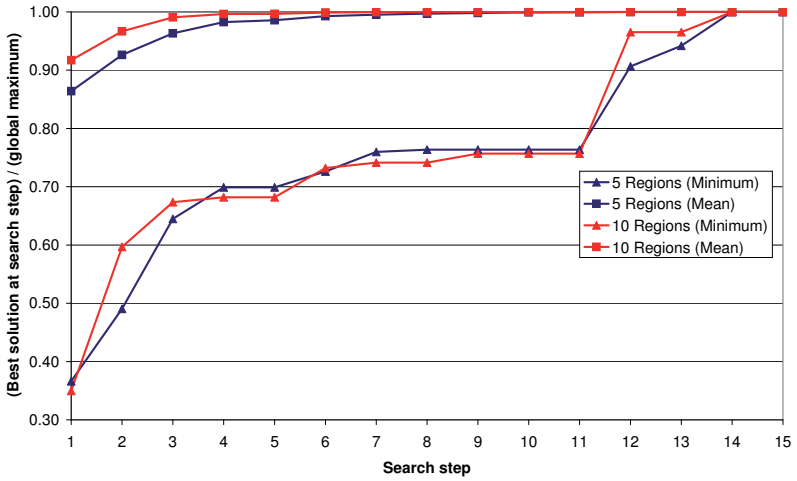


Figure 7.19: Minimum △ and mean □ fraction of the best known solution of the global maximum at a certain search step using pre-sorted search and $N_{\mathcal{S}}$=3 sensor resources.

Comparison of Search Methods

A direct comparison of all search methods is performed by observing the behaviour for increasing numbers of candidate regions $N_{\mathcal{R}}$ as in Fig. 7.20.

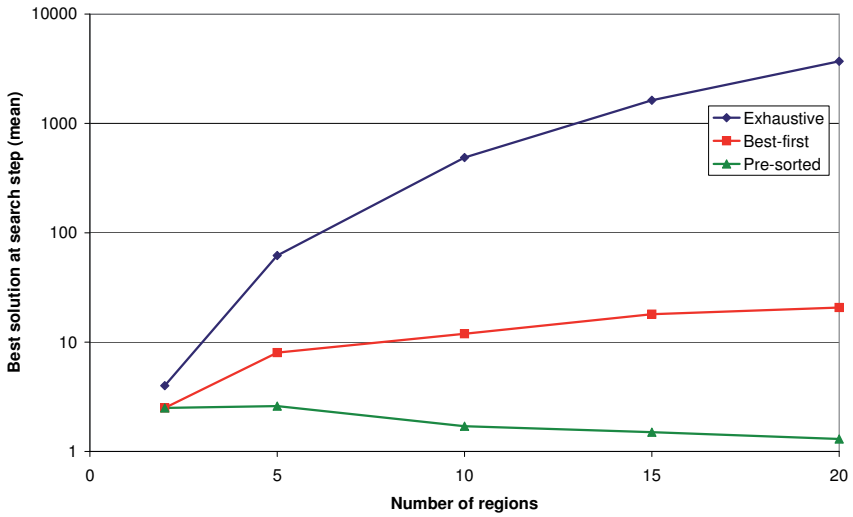


Figure 7.20: Mean number of search steps at which the global maximum is found for the allocation of $N_{\mathcal{S}}$=3 sensor resources and $N_{\mathcal{R}}$ regions using exhaustive search $\diamondsuit$, best-first search $\square$, and pre-sorted search $\triangle$. The ordinate axis is divided logarithmically.

Fig. 7.20 shows that the mean number of search steps necessary to find the global optimum solution is widely different for the discussed search methods.

The search for the actual global maximum is considered inefficient if the best known solution is very close to the global maximum. This convergence is observed for each search method above using the fraction of the best known solution of the global maximum over the initial search steps. The mean values drawn into a single chart can be seen in Fig. 7.21.

From Fig. 7.20 the preference of using either best-first search or pre-sorted search method over an exhaustive search can be seen. Although the pre-sorted search method shows a faster convergence towards the global maximum, the best-first approach is converging faster than could be expected from the mean number of search steps to find the global maximum in Fig. 7.21.

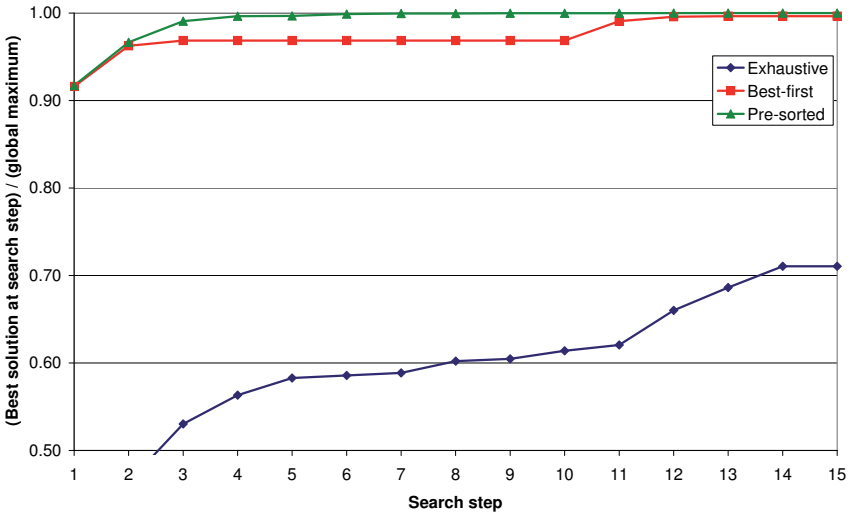


Figure 7.21: Mean fraction of the best known solution of the global maximum at a certain search step using exhaustive search, best-first search, and pre-sorted search for $N_{\mathcal{S}}$=3 sensor resources and $N_{\mathcal{R}}$=10 candidate regions.

7.4 Computational Resource Allocation Heuristics

After the sensor resource allocation is completed, computational resources to classify traffic participants in the acquired sensor data are allocated. Traffic participant classification is a computationally expensive process that in general requires to be run separately on every sensor data set and for different traffic participant types. This leads to a situation where $N_{\mathcal{C}}$ classifier processes compete for a limited amount of computational resources.

Our maximum computational time for classification $t_{\mathcal{C}_{max}}$ before updated sensor data becomes available can be divided between classifier processes by scaling the computational cost of individual classifier processes (cf. section 7.4.1), by scheduling using a priority queue (cf. section 7.4.2), or by a combination of both methods (cf. section 7.4.3). The latter approach is used in our proposed system.

7.4.1 Scaling of Computational Costs

Most classification algorithms allow for scalable computational costs during runtime at the expense of classification quality. Feature-based classification cascades such as the Viola and Jones classifier [52] continually add discriminative features to a stage until a certain minimum true positive rate $P(C|\mathbb{TP})$ is attained while at the same time a given amount of negatives samples is rejected. If more features are used in each stage, the computational cost of the classifier process increases.

For the class of cascaded classifiers, the number of features used in the first stages is most important, as these features are applied on a substantial fraction of the data set. The mean number of features $\overline{N}_f$ applied on negative samples for N stages with a rejection rate r for negative samples in each stage is thus given by

$$\overline{N}_f = \sum_{n=0..N-1} \left(\frac{f(n)}{(1-r_n)^n} \right) \tag{7.26}$$

Generally, the mean feature number decreases for lower admissible true positive rates per stage. In Tab. 7.27 an evaluation of this decrease for a car classifier cascade over the first 10 stages is shown, reducing the minimum true possible rates per stage from 0.997 to 0.985. This causes a decrease of the mean feature number from 16.44 to 6.56 which is a reduction by a factor of 2.51.

$P(C\|\mathbb{TP})_{min}$ per stage	$P(C\|\mathbb{TP})$ at stage 10	Mean feature number
0.997	0.970	16.44
0.995	0.953	10.50
0.990	0.910	9.62
0.985	0.857	6.56

Table 7.27: Mean feature numbers applied on samples for different admissible true positive rates $P(C|\mathbb{TP})$ per stage for car detection using a Viola and Jones cascaded classifier [52].

Due to the exponentially growing denominator in Eq. 7.26 it is possible to increase the minimum true positive rate after the initial stages without significantly affecting the mean feature number and therefore the overall computational cost. We use this method for our trained classifier cascades in sections 5.2.2 to 5.2.5 to reduce the negative effects for classification quality while maintaining an overall low mean feature number.

7.4.2 Queue Scheduling

Scheduling is a form of decision-making, that is responsible for the allocation of scarce resources to optimise resource efficiency (cf. Leung [211]). In operating systems, where process scheduling is one of the main tasks, three levels of scheduling are distinguished according to Stallings [212]:

- long-term scheduling, supervising the admission of processes to the queue,
- mid-term scheduling, supervising the swapping of information in memory, and
- short-term scheduling, supervising the execution order of processes in memory.

While short-term and mid-term scheduling are usually performed by the operation system, long-term scheduling is mostly left to application design. One method used at all levels of scheduling is the placement of all competing processes into a queue. Common examples for queues given by Pruhs *et al.* [213] are

- round-robin queuing, a cycling queue allocating all processes identical time slots before starting a new computation cycle
- fair queuing, which allocates all processes the same fraction of available computational resources (or a fraction proportional to the weight of the process, called weighted fair queuing)
- first-in, first-out queuing, allocating computational resources to the second process only after the first process terminated

In online scheduling, first-in, first-out queues are usually ordered according to their time of arrival. However, it is also possible to order the queue using some heuristics such as shortest job first or highest priority first (cf. Pruhs *et al.* [213]).

Given a maximum available time for classification $t_{\mathcal{C}_{max}}$, preemptive scheduling methods such as round-robin or fair queueing are problematic, as these can end up without any terminated classifier process at $t_{\mathcal{C}_{max}}$. Non-preemptive queuing algorithms such as shortest job first of highest priority first ensure the termination of the preceding classifier process before the next process is started, which is preferable under real-time constraints.

Whether a classifier process running at $t_{\mathcal{C}_{max}}$ is preempted or carried over to the next processing cycle depends on the real-time requirements of the system. In soft real-time systems this is generally decided by the amount of time needed for termination. If this timespan is short, it is considered preferable to delay processing of current sensor data for this process to finish before starting the next cycle. In our proposed system, classifier processes not terminated at $t_{\mathcal{C}_{max}}$ are preempted to reallocate the classifier processes on the updated data.

For scheduling, two properties of classifier processes are important: process priority and process execution time, with ideally both mean execution time and worst-case execution time known. Process priority can be assumed to be given by the partial allocation's utility for every region-sensor allocation in combination with the considered classifier. Process execution time can be obtained by measuring during runtime or by calculation. The latter is often used to determine worst case execution time. For our problem, the worst case is that all samples are analysed by all stages of the classification cascade. However, this

assumption is implausible since the classification cascades are trained to reject a certain fraction of all negative samples at every stage.

In section 7.5.2 both mean execution times and standard deviations of our trained classifier cascades are determined by evaluating the different classification cascades on a set of road-traffic sequences. With the execution time of each process in the queue known, it is possible to calculate the probability of successful termination before $t_{\mathcal{C}_{max}}$. This is effectively a summation of n probability distributions, that is of all mean process execution times $\overline{t_n}$ and the σ after n summations.

If all σ_n are in the same order of magnitude, the overall σ after n summations $\sigma(n)$ can be calculated as the true standard deviation of the mean

$$\sigma(n) = \frac{\bar{\sigma}_n}{\sqrt{n}} \tag{7.27}$$

We use a linearised cumulative distribution function of a normal distribution to determine the probability $P(t_n)$ that a process with an execution time of t_n terminates within Δt as (cf. Fig. 7.22)

$$P(t_n) = \begin{cases} 0.00 & \text{if } \Delta t + 1.75 \cdot \sigma(n) < t_n, \\ 0.50 + \frac{\Delta t}{3.5 \cdot \sigma(n)} & \text{if } \Delta t - 1.75 \cdot \sigma(n) < t_n < \Delta t + 1.75 \cdot \sigma(n), \\ 1.00 & \text{if } \Delta t - 1.75 \cdot \sigma(n) > t_n. \end{cases} \tag{7.28}$$

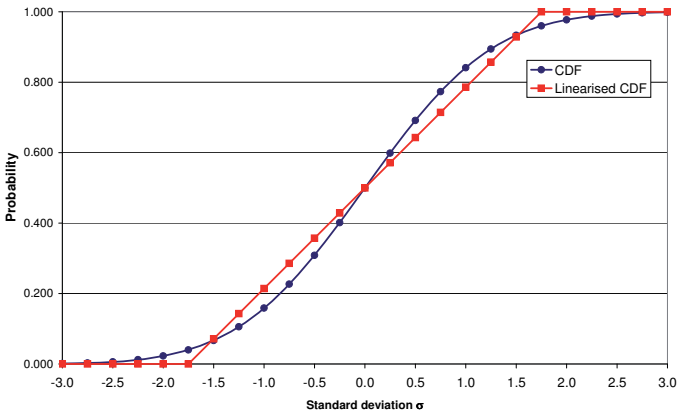


Figure 7.22: Continuous and linearised cumulative distribution function of a normal distribution with zero mean. Linearisation is performed to reduce computational cost during runtime.

An example queue for $N_{\mathcal{C}} = 5$ classifier processes is given in Tab. 7.28.

n	Δt [ms]	$\sigma(n)$ [ms]	$P(t_n)$
1	65	10.00	1.000
2	50	7.07	1.000
3	35	5.77	1.000
4	20	5.00	0.786
5	5	4.47	0.000

Table 7.28: Example queue of $N_{\mathcal{C}}$=5 classifier processes with mean execution times t_n=15 ms and a standard deviation of $\bar{\sigma}$=10 ms. The maximum execution time is $t_{\mathcal{C}_{max}}$=65 ms. The probability $P(t_n)$ to terminate t_n within $t_{\mathcal{C}_{max}}$ is given for every classifier process using Eq. 7.22.

7.4.3 Determination of Classifiers and Priorities

In order to determine the used classifiers and priorities thereof, both scaling of computational costs and prioritisation in the queue must be considered concurrently. For this, the proposed concept translates classifier processes for the same traffic participant but with different classification rates into a base classifier process and a number of virtual classifier upgrade processes. An example for this virtual process partitioning for our trained car classifiers is given in Tab. 7.29 and illustrated in Fig. 7.23. The values for $\Delta P(\mathcal{IP}_4)$ and $t_{\mathcal{C}}$ in Tab. 7.29 are calculated based on Tab. 5.2 and 7.37, assuming a prior probability of $P_{k-1}(\mathcal{IP}_4)$=0.50 and a false positive rate of $P(C|\neg\mathcal{IP}_4) = 0.20$.

Cascade	$P(C\|\mathcal{IP}_4)$	$P(\mathcal{IP}_4\|C)$	$P(\mathcal{IP}_4\|\neg C)$	$\Delta P(\mathcal{IP}_4)$	$t_{\mathcal{C}}$ [ms]
$\mathcal{C}_{4,1}$	0.9069	0.8193	0.1042	0.7151	22.87
$\mathcal{C}_{4,2}$	0.9180	0.8211	0.0930	0.7281	24.37
$\mathcal{C}_{4,3}$	0.9257	0.8223	0.0850	0.7373	25.03

Table 7.29: Values for information measure $\Delta P(\mathcal{IP}_4)$ and process execution time $t_{\mathcal{C}}$ taken from Tab. 7.37. A prior probability of $P_{k-1}(\mathcal{IP}_4)$=0.50 and a false positive rate of $P(C|\neg\mathcal{IP}_4)$=0.20 are assumed.

All base classifier processes are available at the beginning of the allocation process, whereas the virtual classifier upgrade processes $\mathcal{C}_{n,m}$ become available after the base classifier process $\mathcal{C}_{n,1}$ and all lower upgrade processes $\mathcal{C}_{n,2..m-1}$ are allocated. These classifier processes are then inserted either into a queue that guarantees termination of the process running at $t_{\mathcal{C}_{max}}$ or into a non-guaranteed queue that anticipates preemption of any remaining processes in the queue at $t_{\mathcal{C}_{max}}$.

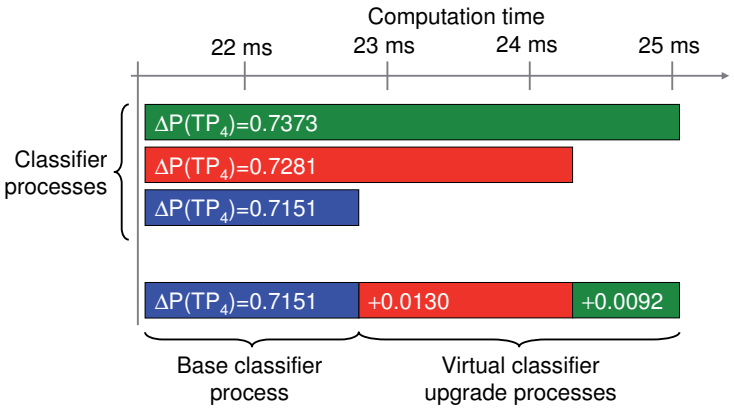


Figure 7.23: Virtual process partitioning for the information measures $\Delta P(\mathcal{TP}_4)$ and process execution times $t_{\mathcal{C}}$ given in Tab. 7.29. The three different classifiers can be partitioned into one base classifier process $\mathcal{C}_{4,1}$ and two virtual classifier upgrade processes $\mathcal{C}_{4,[2,3]}$ for computational resource allocation.

The insertion of classifiers into the queues is then structured as follows

1. From the list of all available virtual classifier processes, the process with the highest utility per second $(\mathcal{U} \cdot s^{-1})$ is appended to the guaranteed queue.
2. The inserted process is removed from the list of available processes, a higher-level upgrade process, if existent, is made available.
3. Repeat steps 1. and 2. while the probability to finish the allocated process is $P(t_n) = 1.00$.
4. Remove all upgrades processes from the list of available classifier processes.
5. From the list of all remaining classifier processes, the process with the highest utility per second probability $(\mathcal{U} \cdot P(t_n) \cdot s^{-1})$ is appended to the preemptive queue.
6. The inserted process is removed from the list of available processes, a higher-level upgrade process, if existent, is made available.
7. Repeat steps 5. and 6. while processes with a probability $P(t_n) > 0.00$ are available.

At this point two queues exist; a guaranteed queue and a preemptive queue. Since the preemptive queue is applied only if the guaranteed queue is processed before the available time $t_{\mathcal{C}_{max}}$ has expired, it is possible to recalculate the preemptive queue after the guaranteed queue is completed using the actual remaining time for the preemptive queue as opposed to the time estimate used for initial queue scheduling.

It is arguable that a recalculation of the preemptive queue renders the initial calculation unnecessary. However, knowledge about sensor-resource combinations that are not

considered in the initial queues enables the system to remove unused sensor-resource combinations from the resource allocation process. This reduces the amount of sensor data to be transferred and processed in our proposed system.

7.4.4 Evaluation of Computational Resource Allocation Heuristics

In this section, the computational resource allocation heuristics for inserting and sorting classifier processes are evaluated using three scheduling methods: maximum (utility per cost) ratio first (MRF), shortest job first (SJF), and maximum utility first (MUF).

Test Conditions

The computational resource allocation heuristics is evaluated using data from $4 \cdot 10^3$ randomly generated classifier process candidate sets. Allocations for $N_{\mathcal{S}}$=3 sensor resources, with $N_{\mathcal{C}}$=3 available classifiers per sensor-region combination and $N_{\mathcal{C}/\mathcal{TP}} = 3$ available classifier scalings per traffic participant type $\mathcal{TP}$, resulting in

$$N_{\mathcal{C}} = N_{\mathcal{S}} \cdot (N_{\mathcal{C}/\mathcal{TP}})^{N_{\mathcal{TP}}} = 3 \cdot 3^3 = 81$$

interdependent processes to be selected and prioritised.

The utility and cost values of all base classifiers and classifier upgrades are generated randomly with a mean value $\bar{x} = 10$ and a standard deviation $\sigma_x = 2$. Negative values for x are discarded, as neither negative utility nor negative cost is possible. The actual execution time of the processes is simulated by superimposing a Gaussian noise with $\sigma(x) = 0.2 \cdot \bar{x}_{est}$ onto the estimated cost x_{est}. The maximum available computational cost is chosen to be $x_{C_{max}} = 65$ ms.

Avoid Decision

For the given test conditions, the avoidance of making a decision is equivalent to a static queue. Considering our test conditions of 4,000 randomly generated classifier sets, a predetermined queue is equivalent to a random decision. An optimum predetermined queue for our trained classifiers is given in section 7.5.2.

Random Decision

The classifier queue can be generated randomly, that is both the scaling and the queue scheduling are purely random processes. Under our test-conditions, the mean utility sum of all classifiers $\bar{x}$ and the mean number of classifiers processed $\bar{n}_{\mathcal{C}}$ processed within $t_{\mathcal{C}_{max}}$ using a random queue are

$$\bar{x} = 55.31, \quad \bar{n}_{\mathcal{C}} = 3.12$$

with a standard error of the mean of $SE_{\bar{x}} = 0.20$ and $SE_{\bar{n}_{\mathcal{C}}} = 0.01$.

Generation of a Static Queue

The static queue is determined at the beginning and not changed until $t_{\mathcal{C}_{max}}$ is expired, at which point a new static queue is determined using the new estimated utility and cost values.

		Sorting		
		MRF	SJF	MUF
Inserting	MRF	71.79	67.76	72.93
	SJF	67.95	64.91	68.84
	MUF	67.38	62.94	68.76

Table 7.30: Mean utility sum $\bar{x}$ of all classifiers processed within $t_{\mathcal{C}_{max}}$ using a static queue. Used scheduling methods are maximum (utility per cost) ratio first (MRF), shortest job first (SJF), maximum utility first (MUF). Standard error of the mean is $SE_{\bar{x}} \leq 0.25$.

For static queues the combination of a MRF method for inserting and a MUF method for sorting the combined queues shows the highest mean utility sum.

		Sorting		
		MRF	SJF	MUF
Inserting	MRF	4.08	4.06	4.03
	SJF	4.17	4.18	4.17
	MUF	3.83	3.79	3.81

Table 7.31: Mean number of classifiers $\bar{n}_{\mathcal{C}}$ processed within $t_{\mathcal{C}_{max}}$ using a static queue. Used scheduling methods are maximum (utility per cost) ratio first (MRF), shortest job first (SJF), maximum utility first (MUF). Standard error of the mean number of classifiers is $SE_{\bar{n}_{\mathcal{C}}} \leq 0.02$.

It can be seen in Tab. 7.31 that the methods used for sorting do not have a measurable impact upon the mean number of processed classifiers considering a standard error of $SE_{\bar{x}} \leq 0.02$. However, the insertion method greatly influences the mean number of processed classifiers, with SJF exhibiting the highest classifier throughput.

Queue Recalculation after each Classifier Process

As an alternative to a static queue, the queue is recalculated after each termination of a classifier process using the updated remaining estimated utility and cost values. As for the static queue, the mean utility sum of all classifiers and the mean number of classifiers is given in Tab. 7.32 and 7.33 below.

		Sorting		
		MRF	SJF	MUF
Inserting	MRF	73.18	67.64	75.34
	SJF	68.17	64.79	69.67
	MUF	69.76	65.19	71.15

Table 7.32: Mean utility sum $\bar{x}$ of all classifiers processed within $t_{\mathcal{C}_{max}}$. Used scheduling methods are maximum (utility per cost) ratio first (MRF), shortest job first (SJF), maximum utility first (MUF). Standard error of the mean is $SE_{\bar{x}} \leq 0.34$.

For static queues, the combination of a MRF method for inserting and a MUF for sorting the combined queues shows the highest mean utility sum for iteratively recalculated queue scheduling.

		Sorting		
		MRF	SJF	MUF
Inserting	MRF	4.04	4.63	4.00
	SJF	5.80	6.39	4.74
	MUF	4.13	4.76	3.80

Table 7.33: Mean number of classifiers $\bar{n}_{\mathcal{C}}$ processed within $t_{\mathcal{C}_{max}}$. Used scheduling methods are maximum (utility per cost) ratio first (MRF), shortest job first (SJF), maximum utility first (MUF). Standard error of the mean number of classifiers is $SE_{\bar{n}_{\mathcal{C}}} \leq 0.03$.

As for static queues, using SJF is the best method for both inserting and sorting when using queue recalculation. The impact of using SJF as a sorting algorithm is different, as it significantly increases the mean number of processed algorithms for all insertion methods. This is in contrast to the static queue, where the used sorting method does not have an observable impact.

Discussion of Computational Resource Evaluation

From the above evaluation of queue scheduling methods using our classifier concept the following can be inferred:

- The use of shortest job first for insertion and sorting provides the highest mean number of classifiers $\bar{n}_{\mathcal{C}}$ processed within $t_{\mathcal{C}_{max}}$.
- The combination of maximum utility per cost ratio for insertion and maximum utility for sorting provides the highest overall mean utility $\bar{x}$within $t_{\mathcal{C}_{max}}$.
- The gain in overall mean utility within $t_{\mathcal{C}_{max}}$ by queue recalculation is small ($\leq 3.5\%$).

While the first observation of attaining highest classifier throughput by using the shortest job first method is expected and also described by Pruhs *et al.* [213], the second inference from our evaluation is more interesting.

For our given problem, the combination of maximum utility per cost ratio for insertion and maximum utility for sorting showed to provide the maximum utility within $t_{\mathcal{C}_{max}}$. Using a maximum utility per cost ratio for sorting leads to classifiers with a higher utility being processed later in the queue. As high utility is often paired with higher computational cost, this increases the probability that a high-utility classifier process is preempted, thus reducing overall utility. Sorting towards a decreasing expected utility value can mitigate this problem.

The third observation of only a small gain ($\leq 3.5\%$) in overall mean utility for queue recalculation is interesting, as the optimum queue for the remaining time is determined iteratively. This method minimises the probability of classifiers to be preempted at $t_{\mathcal{C}_{max}}$. Further investigation of this effect shows two reasons for this small difference: overall small number of preempted processes and comparably low utility of preempted processes.

The overall number of processes that suffer from preemption is very small. In Tab. 7.34 the number of successfully terminated processes and preempted processes for two test runs using an (MRF/MUF) scheduling are shown.

	Terminated $\mathcal{C}$	Preempted $\mathcal{C}$	Preemption ratio
Static queue	14961	1160	0.078
Queue recalculation	15596	388	0.025

Table 7.34: Number of successfully terminated processes and preempted processes for two test runs using (MRF/MUF) scheduling.

It can be seen from Tab. 7.34 that the preemption rate for a static queue is approximately three times higher than using queue recalculation. However, only 7.8% of the processes get preempted while the other processes terminate successfully.

The number of preempted processes would suggest a 7.8% - 2.5% = 5.3% difference

between static queues and queue recalculation. The actual difference is even smaller, as processes that get preempted are usually at the back of the processing queue. Using a maximum utility first (MUF) sorting for scheduling, the utility of the processes show a considerable decrease. The mean utility value for preempted processes is compared against the mean overall value in Tab. 7.35.

	mean overall process utility	mean preempted process utility
static queue	18.10	11.79
queue recalculation	18.83	12.68

Table 7.35: Mean utility value for all processes and preempted processes.

Combining the findings from Tab. 7.34 and Tab. 7.35 it can be seen that

$$(0.078 - 0.025) \cdot \frac{11.79}{18.10} = 0.053 \cdot 0.651 = 0.0345$$

which then explains the difference of $\leq$3.5% between the use of static queues and queue recalculation.

7.5 Evaluation of Contextual Resource Allocation

Earlier in this chapter an evaluation of the decision making concepts in section 7.2.3 and resource allocation heuristics in sections 7.3.5 and 7.4.4 is given. In this section, our contextual resource allocation is evaluated using the road traffic sequences recorded with our test vehicle. Corresponding to our partition of resource allocation into sensor allocation and computational resource allocation, the former is evaluated in section 7.5.1 and the latter is evaluated in section 7.5.2. The resulting allocations for our test sequences are presented and analysed for a subset of every sequence in section 7.5.3.

7.5.1 Evaluation of Contextual Sensor Resource Allocation

In order to evaluate our proposed contextual sensor resource allocation system three video sensor models are used. The sensor allocation is then tested on three sequences and the resulting scores are presented.

Sensor Models for Evaluation

For our evaluation, three video sensor models defined in Tab. 7.36 are used. All video images are based upon a 640×480 px video frame acquired by a fixed camera in our test vehicle. The video frame is then upscaled for selected regions ($\mathcal{S}_1$, $\mathcal{S}_2$) and downscaled ($\mathcal{S}_3$) to simulate video cameras with different angular resolutions. This process is presented in section 3.2. For our simulated PTZ camera $\mathcal{S}_2$ a maximum gaze shift velocity of 360°/s equal to the PTZ camera mounted on the test vehicle's roof is imposed.

	Cropping ($\mathcal{S}_1$)	PTZ ($\mathcal{S}_2$)	Wide-angle ($\mathcal{S}_3$)
Region resolution [px]	320×240	320×240	$\leq$ 320×240
Region size [°]	10×7.5	10×7.5	40×30
Angular resolution [px/°]	32.0	32.0	8.0
Gaze shift velocity [°/s]	not applicable	360	not applicable

Table 7.36: Video sensor models used for sensor resource allocation evaluation. $\mathcal{S}_1$ crops a candidate region from a 1280 × 960 pixel image. $\mathcal{S}_2$ is a simulated PTZ camera, with a maximum gaze shift velocity of 360°/s. $\mathcal{S}_3$ is a low-level wide angle video image.

Apart from our basic sensors $\mathcal{S}_1$ to $\mathcal{S}_3$, the concept of virtual sensors is introduced in section 7.3.4. According to this concept, four virtual sensors $\mathcal{S}_{12}, \mathcal{S}_{13}, \mathcal{S}_{23}$ and $\mathcal{S}_{123}$ exist.

For the chosen sensor sensor models, all sensor data is obtained from the same fixed camera, ruling out additional or emerging modalities. Therefore the observation of a single region with multiple sensors does not increase the quality of sensor data. This can be modelled by equating the virtual sensor with the basic sensor exhibiting both the highest resolution and the farthest-reaching constraints such as region size and gaze shift velocity. For our sensors, the following equivalents can be used

$$\mathcal{S}_{13} \equiv \mathcal{S}_1, \quad (\mathcal{S}_{12}, \mathcal{S}_{23}, \mathcal{S}_{123}) \equiv \mathcal{S}_2$$

Resulting Scores for Test Sequences

Our contextual sensor resource allocation scheme is evaluated using three test sequences acquired on a traffic calmed road (TRC), an urban road (URB), and a motorway (MWY). As a quality indicator for the sensor-region allocations, the criticality score η is determined using Eq. 7.15.

The resulting scores are shown in Fig. 7.24. There it can be seen that the use of uncertainty information for resource allocation increases the criticality of observed regions and

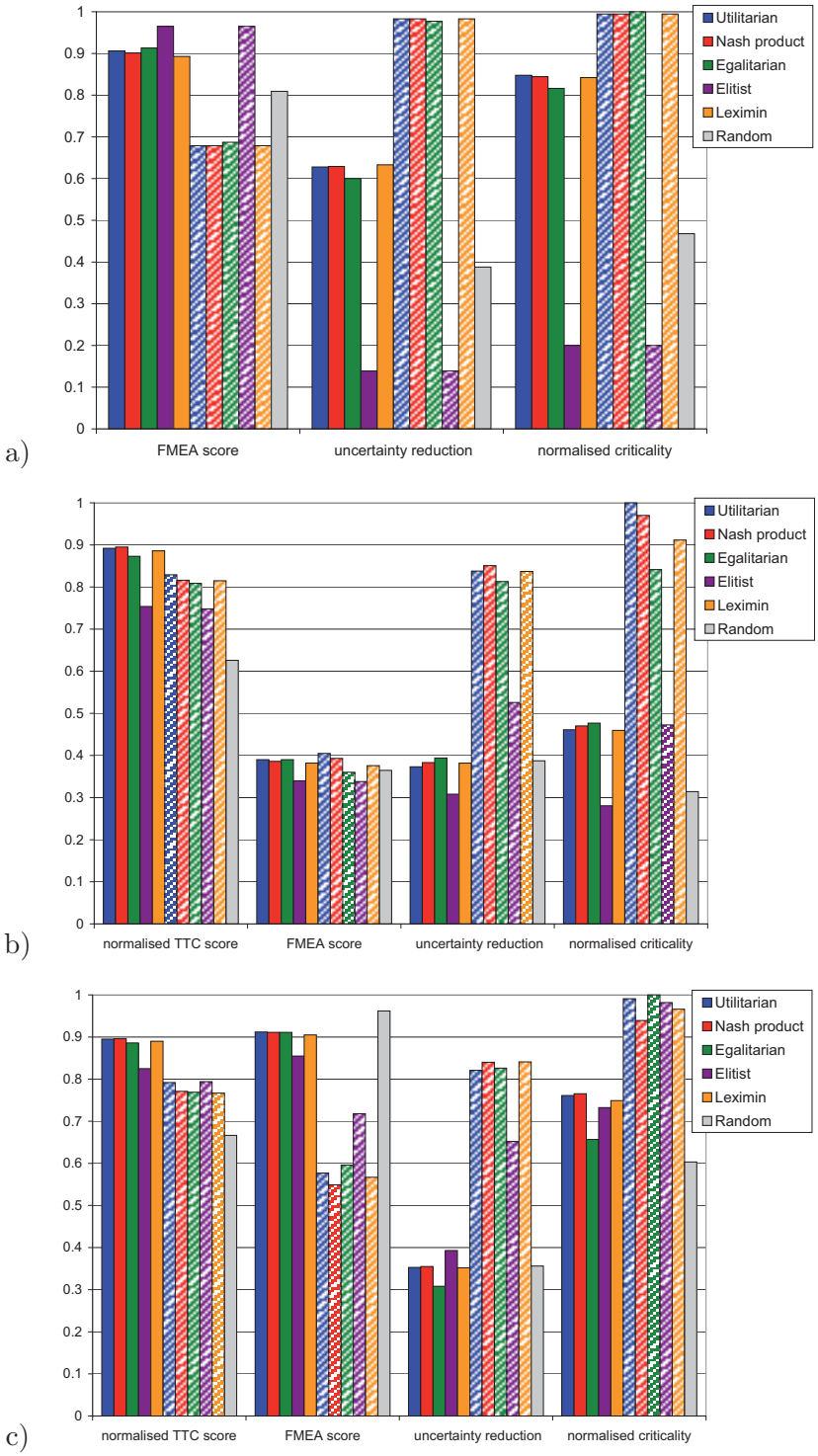


Figure 7.24: Resulting mean TTC score (where available), mean FMEA score, mean uncertainty reduction, and mean overall criticality η for three sequences using our proposed system. Solid bars indicate that no uncertainty information is considered in the combined utility, which is the case for hatched bars. Gray bars indicate a random selection among the candidate regions. Figure a) shows the results for the traffic calmed road sequence (TRC), b) for the urban road sequence (URB), and c) for the motorway sequence (MWY).

thereby sensor utilisation. The use of uncertainty information is proposed, as it facilitates a minimisation of uncertainty about the ego-vehicle's environment. If uncertainty is not considered, the criticality values in Fig. 7.24 show to be considerably lower, resulting in a reduced sensor utilisation.

As for the evaluation of the candidate regions' criticality in Fig. 7.11 on p. 189, Pareto efficient utility concepts such as Utilitarian, Nash product, and Leximin utility concepts show to select regions with a high criticality.

7.5.2 Evaluation of Contextual Computational Resource Allocation

In order to evaluate our proposed contextual computational resource allocation system, the execution times for our trained classifier cascades are determined. The computational resource allocation is then tested using the trained classifiers' execution times.

Classifier Execution Times

The execution times of our traffic participant classifiers are measured using a sequence of 375 road traffic images at a resolution of 320×240 px. The statistical properties of the classifier execution times are given in Tab. 7.37.

The measured classifiers' mean execution times μ in Tab. 7.37 and their respective standard deviations σ are drawn against the classifiers' true positive rates in Fig. 7.25.

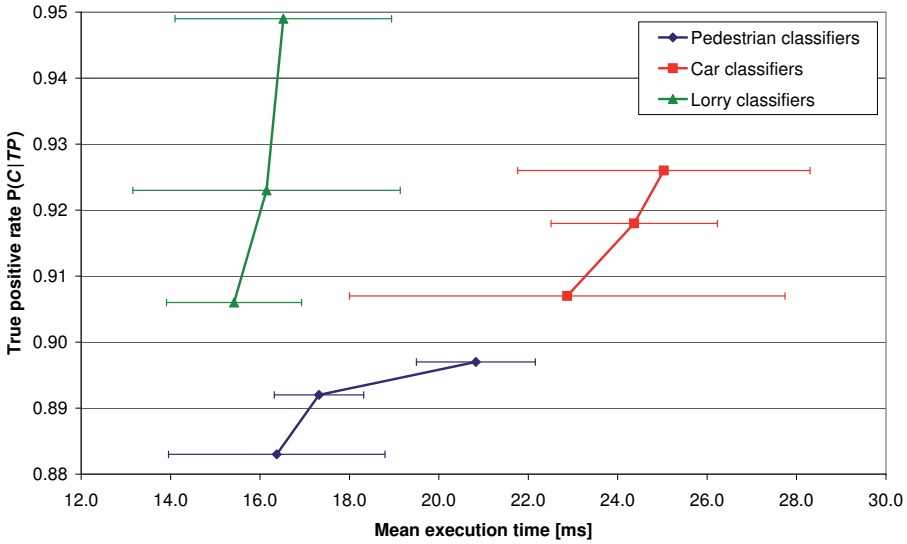


Figure 7.25: Measured mean execution times μ with standard deviations σ for our traffic participant classifier cascades from Tab. 7.37 with corresponding true positive rates $P(C|\mathcal{TP})$.

$P(C\|\mathcal{TP}_1)$	0.883	0.892	0.897
Minimum [ms]	14.49	15.55	19.17
Median [ms]	15.92	17.19	20.54
Maximum [ms]	38.42	21.87	27.53
Mean (μ) [ms]	16.38	17.32	20.83
SD (σ) [ms]	2.42	1.00	1.33
SE_μ [ms]	0.12	0.05	0.07

$P(C\|\mathcal{TP}_4)$	0.907	0.918	0.926
Minimum [ms]	15.68	20.88	20.80
Median [ms]	21.57	24.06	24.33
Maximum [ms]	50.51	33.98	41.85
Mean (μ) [ms]	22.87	24.37	25.03
SD (σ) [ms]	4.87	1.86	3.27
SE_μ [ms]	0.17	0.40	0.22

$P(C\|\mathcal{TP}_5)$	0.906	0.923	0.949
Minimum [ms]	13.97	14.15	14.21
Median [ms]	14.95	15.73	16.16
Maximum [ms]	25.86	57.82	40.47
Mean (μ) [ms]	15.42	16.15	16.52
SD (σ) [ms]	1.51	2.99	2.42
SE_μ [ms]	0.08	0.15	0.12

Table 7.37: Measured execution times of trained classifier cascades for pedestrians ($\mathcal{C}_{1,1}$ to $\mathcal{C}_{1,3}$), cars ($\mathcal{C}_{4,1}$ to $\mathcal{C}_{4,3}$), and lorries ($\mathcal{C}_{5,1}$ to $\mathcal{C}_{5,3}$) on a 320×240 px image sequence. Mean execution time μ and standard deviation σ are relevant for our resource allocation algorithm. The standard error of the mean SE_μ is given as an indicator of the mean's accuracy.

In Fig. 7.25 two notable properties can be observed. First, the increase in computational time varies between different traffic participant types. The mean execution time for pedestrian classifiers increases significantly with an increasing true positive rates, which is an effect that is less prominent for car classifiers and lorry classifiers. Second, standard deviations from the mean execution times are considerable in the range of $\sigma = [1.00, 4.87]$ ms, corresponding to a relative standard deviation between 5.8% and 21.2%. Large standard deviations complicate the determination of an optimal processing queue, in particular if no queue recalculation is performed.

Resulting Performance for Trained Classifiers

The mean execution times and standard deviations of our trained classifier processes presented in Tab. 7.37 are used to evaluate our contextual resource allocation system. Classifier utility $\mathcal{U}_{\mathcal{C}}$ is defined as the classifier's information gain measures $\Delta P(\mathcal{IP}_n)$ dependent upon the observed region $\mathcal{R}_\iota$ and classifier $\mathcal{C}_{n,m}$.

$$\mathcal{U}_{\mathcal{C}}(\mathcal{R}_\iota, \mathcal{C}_{n,m}) = \Delta P(\mathcal{IP}_n | \mathcal{R}_\iota, \mathcal{C}_{n,m}) \tag{7.29}$$

The maximum available time $t_{\mathcal{C}_{max}}$ is assumed as 65 ms, corresponding to 15 fps. The resulting mean utility sum of the queue using our computational resource allocation is given in Tab. 7.38 and the mean number of terminated classifier processes is given in Tab. 7.39.

		Sorting		
		MRF	SJF	MUF
Inserting	MRF	0.995	0.965	1.000
	SJF	0.792	0.802	0.785
	MUF	0.957	0.930	0.977

Table 7.38: Normalised mean utility sum $\bar{\mathcal{U}}$ of trained classifiers processed within 65 ms. Used scheduling methods are maximum (utility per cost) ratio first (MRF), shortest job first (SJF), maximum utility first (MUF). Standard error of the mean is $SE_{\bar{x}} \leq 0.009$.

		Sorting		
		MRF	SJF	MUF
Inserting	MRF	2.39	2.48	2.30
	SJF	2.83	3.06	2.71
	MUF	2.29	2.29	2.26

Table 7.39: Mean number of trained classifiers $\bar{n}_{\mathcal{C}}$ processed within 65 ms. Used scheduling methods are maximum (utility per cost) ratio first (MRF), shortest job first (SJF), maximum utility first (MUF). Standard error of the mean number of classifiers is $SE_{\bar{n}_{\mathcal{C}}} \leq 0.01$.

Using a random queue generation, the normalised mean utility $\bar{\mathcal{U}}$ and mean number of terminated classifier processes $\bar{n}_{\mathcal{C}}$ are

$$\bar{\mathcal{U}} = 0.644, \quad \bar{n}_{\mathcal{C}} = 2.62$$

Optimum predetermined Classifier Queue

Using both the information gain measures $\Delta P(\mathcal{IP})$ and computation times $t_{\mathcal{C}}$ of our trained classifiers, an optimum predetermined classifier queue for all road types $\mathcal{RT}_m$ is determined. In Tab. 7.40 the Utility per time ratios of our trained classifiers are determined assuming $P(\mathcal{IP})$=0.50 and $P(C|\neg\mathcal{IP})$=0.20.

$$\mathcal{U}_{\mathcal{C}}(\mathcal{IP}_n) = \Delta P(\mathcal{IP}_n) \sum_m \left(\widehat{s}(\mathcal{IP}_n, \mathcal{RT}_m) \cdot \frac{P(\mathcal{IP}_n, \mathcal{RT}_m)}{\sum_\iota P(\mathcal{IP}_n, \mathcal{RT}_\iota)} \right) \tag{7.30}$$

resulting in

$$\mathcal{U}_{\mathcal{C}}(\mathcal{IP}_1) = \Delta P(\mathcal{IP}_1) \cdot 1.125,$$

$$\mathcal{U}_{\mathcal{C}}(\mathcal{IP}_4) = \Delta P(\mathcal{IP}_4) \cdot 0.613,$$

$$\mathcal{U}_{\mathcal{C}}(\mathcal{IP}_5) = \Delta P(\mathcal{IP}_5) \cdot 0.802$$

Classifier	$P(C\|\mathcal{IP})$	$P(\mathcal{IP}\|C)$	$P(\mathcal{IP}\|\neg C)$	$\Delta P(\mathcal{IP})$	$\mathcal{U}_{\mathcal{C}}$	$t_{\mathcal{C}}$ [ms]	$\frac{\mathcal{U}_{\mathcal{C}}}{t_{\mathcal{C}}}$ $[\frac{1}{s}]$
$\mathcal{C}_{1,1}$	0.883	0.8153	0.1276	0.6877	0.7737	16.36	47.30
$\mathcal{C}_{1,2}$	0.892	0.8168	0.1189	+0.0102	+0.0118	+0.96	11.91
$\mathcal{C}_{1,3}$	0.897	0.8177	0.1141	+0.0057	+0.0064	+3.51	1.83
$\mathcal{C}_{4,1}$	0.907	0.8193	0.1042	0.7151	0.4384	22.87	19.17
$\mathcal{C}_{4,2}$	0.918	0.8211	0.0930	+0.0130	+0.0080	+1.50	5.31
$\mathcal{C}_{4,3}$	0.926	0.8223	0.0850	+0.0092	+0.0056	+0.66	8.55
$\mathcal{C}_{5,1}$	0.906	0.8192	0.1051	0.7140	0.5726	15.42	37.13
$\mathcal{C}_{5,2}$	0.923	0.8219	0.0878	+0.0201	+0.0161	+0.73	22.06
$\mathcal{C}_{5,3}$	0.949	0.8259	0.0599	+0.0319	+0.0256	+0.37	69.15

Table 7.40: Computation times and utility per time ratios for base classifier processes and virtual classifier upgrade processes derived from our trained classifier cascades.

Using the $\mathcal{U}$/s ratios given in Tab. 7.40, an optimum predetermined classifier queue is given in Tab. 7.41.

Inserting	$t_\mathcal{C}$ [ms]	$\frac{\mathcal{U}_\mathcal{C}}{t_\mathcal{C}}$ $[\frac{1}{s}]$	$\sum t_\mathcal{C}$	$P(t_\mathcal{C})$
$\mathcal{C}_{1,1}$	16.36	47.30	16.36	1.000
$\mathcal{C}_{5,1}$	15.42	37.13	31.78	1.000
$\mathcal{C}_{5,2}$	+0.73	22.06	32.51	1.000
$\mathcal{C}_{5,3}$	+0.37	69.15	32.88	1.000
$\mathcal{C}_{4,1}$	22.87	19.17	55.75	1.000
$\mathcal{C}_{1,2}$	+0.96	11.91	56.71	1.000
$\mathcal{C}_{4,2}$	+1.50	5.31	58.21	0.809
$\mathcal{C}_{4,3}$	+0.66	8.55	58.87	0.709
$\mathcal{C}_{1,3}$	+3.51	1.83	62.38	0.034
Sorting	$t_\mathcal{C}$ [ms]	$\mathcal{U}_\mathcal{C}$	$\sum t_\mathcal{C}$	$P(t_\mathcal{C})$
$\mathcal{C}_{4,1}$	22.87	0.8045	22.87	1.000
$\mathcal{C}_{5,3}$	16.52	0.6143	39.39	1.000
$\mathcal{C}_{1,2}$	17.32	0.4278	56.71	1.000

Table 7.41: Optimum predetermined classifier queue using the computation times and utility per time ratios given in Tab. 7.40.

7.5.3 Resulting Allocations for Test Sequences

In order to examine the resulting allocations for three test sequences (Trc, Urb, and Mwy), a subset of six short frame sequences from the complete sequence is shown in appendix B to represent both operation with auxiliary traffic participant detection results and operation under the adverse influence of false positive detections.

A discussion of the resulting allocation for the test sequence is given in section 7.6.3.

7.6 Discussion of Contextual Resource Allocation

In this chapter the contextual resource allocation of our proposed system is presented. This section provides a discussion of methods and concepts. First, the severity determination concept and possible amendments to the presented method are discussed in section 7.6.1. Second, the presented resource allocation concept is discussed in section 7.6.2. Third, the evaluated resulting allocations for the test sequences in section 7.5 are discussed in section 7.6.3.

7.6.1 Severity Determination

A utilitarian concept to determine the expected severity of an accident with another traffic participant on a given road type is used. This choice is argued in sections 2.3.1 and 7.2.2 to

be morally problematic but applicable in a driver assistance system context as opposed to autonomous driving. Apart from the severity determination concept used in our proposed system, three potential amendments are suggested.

First, it is arguable to use only the probability of a lethal injury as a severity indicator. This accounts for the non-negotiability of life as well as rendering the questionable use of socio-economic costs obsolete.

Second, unmotorised traffic participants such as pedestrians and bicycles are not dangerous to any other traffic participant types. Following the Fourth Geneva Convention [77] these traffic participant groups would have to be treated as civilians, guaranteeing the highest severity to any region with detected pedestrians or bicycles.

Third, the number of traffic participants in a region can be obscured towards the decision making instance. This method of using *veil of ignorance* is proposed by Rawls [214], again accounting for the non-negotiability of life. A region with multiple pedestrians would therefore not be preferred to a region with a single pedestrian.

The above amendments are inherently consistent and parts of them can already be found in our proposed severity determination concept.

First, both light injuries and severe injuries are considered in our system. However lethal injuries are assigned a 300 times higher and 13 times higher severity score respectively. In practice, this raises the influence of lethal injuries on the severity score above all other categories.

Second, the mean number of injuries in Tab. 7.7 reflects the high vulnerability of pedestrians and bicycles. Although not accounting for the civilian status, this ensures a high severity for these traffic participant groups. This is only valid for state of the art technology, where the vulnerability of unmotorised traffic participants is high. If, by advances in automotive safety technology, the vulnerability of a pedestrian becomes comparable to the vulnerability of a person in a car, the status difference still remains. In that case, the status difference originates from actively consenting to participate in road traffic (e.g. motorised traffic participants) and being forced to participate in road traffic (e.g. pedestrian crossing a road). An acceptable differentiation of the degree of consent and thus the status of a traffic participant presents future work in the field of applied ethics.

Third, the number of traffic participants is partly obscured in our system by transferring only the probability that at least one traffic participant of a certain type exists

inside a candidate region. For example a probability of $P(\mathcal{TP}_n) = 0.96$ can originate from a single detection with $P(\mathcal{TP}_n|C) = 0.96$ or from two detections with $P(\mathcal{TP}_n|C) = 0.80$.

7.6.2 Resource Allocation Concept

The discussion of the proposed resource allocation concept is divided into the determination of the optimal utility concept, the possible allocation of weights to objectives, and the novelty aspect of our resource allocation.

Ranking of Utility Concepts for Resource Allocation

Based upon the robustness of criticality estimation over all test sequences given in Fig. 7.24, a ranking of utility concepts is established. For both the sum and the product of all normalised criticality scores c, the resulting ranking of utility concepts is identical (cf. Tab. 7.42).

Rank	Utility concept	$\sum_n(c_n)$	$\prod_n(c_n)$
1	Utilitarian utility	2.99	0.986
2	Nash product utility	2.90	0.906
3	Leximin utility	2.87	0.876
4	Egalitarian utility	2.84	0.842
5	Elitist utility	1.65	0.093
6	Random allocation	1.39	0.089

Table 7.42: Ranking of utility concepts for contextual resource allocation based upon the resulting criticality scores given in Fig. 7.24. Ranks are determined using both the sum and the product of the three criticality scores, which lead to the same ranking.

As a conclusion from Tab. 7.42 we propose the use of a utilitarian utility concept for contextual resource allocation.

Allocation of Weights to Objectives

Assigning different weights to different objectives is an obvious extension to our proposed system. This can either take the form of a predetermined static weight for every objective or a dynamic weight corresponding to the individual objective's current *drive strength*, a concept presented by Seara and Schmidt [88, 89].

An optimal static weight distribution can be obtained by testing different weight distributions on test sequences, evaluating the allocation performance. A concept to derive

dynamic weights is proposed in section 6.3.1, considering the variance of the traffic participant probabilities as an indicator of the dependent objective's current credibilities.

Our proposed system refrains from allocating weights to objectives, as its contribution lies in the determination of an optimal utility concept for resource allocation. Evaluation of a possible increase in quality by using objective weights therefore presents future work.

Novelty Aspect

Our proposed resource allocation system presents an original contribution based on four properties discussed below.

First, the resource allocation problem constituted by an active vision system is formalised. For this, the formalisation concept presented by Chevaleyre *et al.* [83] is used. The active vision concepts discussed in section 2.4 of the literature review do not provide a formalised problem statement, which in turn impedes an optimum solution.

Second, a Pareto efficient decision making process is used to determine the optimum resource allocation. It can be seen from Tab. 2.5 that of all reviewed active vision concepts only the utility based concept presented by Seara and Schmidt [88, 89] is also formally Pareto efficient, but lacks the capability to operate in real-time. Other methods such as the integrated model by Navalpakkam and Itti [137], goal-directed search by Frintrop [103], and contextual guidance model by Torralba *et al.* [146] appear to be Pareto efficient. However this property is neither explicitly intended, nor claimed in the respective publications.

Third, as opposed to a single bottom-up and a single top-down cue, a total of five independent objectives to determine the relevance of a candidate region is used. Moreover, the objectives include prior knowledge about accident severities dependent upon both road type and traffic participant type. This information is important for a prioritisation of both vulnerable and dangerous traffic participants in the environment.

Fourth, the complexity of our decision making system is reduced by selecting a limited set of candidate regions, using a rank-degradation method for sensor-region allocation and introduce the concept of virtual classifier upgrade processes for classifier queue generation.

7.6.3 Discussion of Resulting Allocations for Test Sequences

An evaluation of the resulting allocations for our test sequences is given in section 7.5. There, the overall number of true positives and false positives for detectors and allocated

classifiers in Tab. B.1 to B.3 allows a quantitative observation: For scenes with auxiliary detection results, the number of both true positives, but also false positives generally increases. For scenes with adverse detection results, the number of true positives increases and the number of false positives generally decreases. An exception for the decrease of false positives are the classifier results in the URB sequence. The large number of false positives indicates that the classifier cascades for an automotive system have to be trained using a larger dataset in order to increase robustness.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

In this thesis an original resource allocation concept for automotive vision systems is proposed. We claim that the presented system is capable of efficiently allocating both sensor resources, and computational resources towards relevant regions in the environment. This claim is substantiated by an evaluation using multi-sensor data acquired by a test vehicle provided by Audi AG.

Our proposed system is organised in five levels of abstraction. This layered architecture ensures that the amount of processed and transferred data decreases as the level of abstraction increases. The reduction of processed data lowers the computational demands on the vehicle's electronic control units and the reduction of transferred data reduces the load of the vehicle's bus system. In order to minimise the latency caused by serial processing over multiple levels, processes within the same levels are run in parallel. In addition, semantic information is made available to driver assistance systems in the third out of five levels, with both sensor level, and data level processes designed to be computationally inexpensive.

Used data processing methods are in part proven algorithms such as the Viola and Jones detector [52], but also novel methods, e.g. PCS motion estimation proposed by Matzka *et al.* [215]. Apart from the methods used in the proposed system, the generation of efficient scan-patterns for spin image based classifiers (cf. section 5.3) is investigated for use in future systems.

The central contribution of this thesis is the formalisation and evaluation of the decision making process required for resource allocation, extending existing active vision systems discussed in section 2.4. Our proposed system is novel in the respect that it combines a formal, Pareto efficient decision making method with bottom-up and top-down information acquired using low-resolution data. This is in contrast to methods presented in the literature selecting regions of interest from high-resolution data.

In our evaluation we show that a multi-objective optimisation of five independent objectives

- regions with vulnerable or dangerous traffic-participants Ω_1,
- salient regions Ω_2,
- regions with critical time-to-collision values Ω_3,
- regions for which observation results in a high uncertainty reduction Ω_4, and
- regions that can be observed and processed in the available time Ω_5.

allows determination of candidate regions, allocation of sensors to regions, and allocation of classifier processes to high-resolution sensor data. These resource allocation processes are efficient in two respects. First, it is shown that the use of heuristics minimises the computational requirements of the allocation process itself. Second, the quality of the determined allocations, evaluated using a criticality score, is significantly better than using a static allocation or a random allocation.

We show that the use of a Utilitarian utility concept $\mathcal{U}^u$ for decision making in a contextual resource allocation is preferable to other Pareto efficient methods such as Nash product $\mathcal{U}^\times$, or Leximin ordering $\mathcal{U}^\lambda$ concept. All of the former methods are in turn preferable to methods not guaranteeing Pareto efficiency such as Egalitarian utility $\mathcal{U}^e$, Elitist utility $\mathcal{U}^\$$, and random decisions.

The high complexity of determining the best sensor-region combination is countered in our system by using a pre-sorted search heuristic. The proposed method determines the global optimum in less than three search steps on average. For computational resource allocation, we propose a virtual classifier upgrade concept, inserting classifiers with the highest utility per computation time into the classifier process queue, and assigning priorities using a highest utility first method.

All proposed methods and the complete active vision system are tested using both synthetic data and multi-sensor road traffic test sequences acquired by the test vehicle

provided by Audi AG. Considering the results of our investigations, we propose to extend the presented system as described in section 8.2 on future work.

8.2 Future Work

In this thesis, four aspects of future work are identified. In the following, possible extensions to the existing system are proposed. The proposed extensions are given in descending order of both priority and feasibility.

Training of Robust Classifiers

In this thesis, a set of two detector cascades and six classifier cascades are presented. The cascades are trained using 70 to 750 independent (i.e. not mirrored or rotated) positive samples for pedestrians, cars, and lorries. Also, no positive samples of bicycles and motorcycles are used. The classifiers show good results on the test sets and for two test sequences (TRC and MWY), but also a significant number of false positives for our URB test sequence.

In order to obtain a set of robust classifier cascades with comparable results for all road scenes, the number of independent positive and negative samples have to be increased significantly. Positive samples of bicycles and motorcycles must be included for both detector cascades and a set of classifier cascades for each traffic participant type. Moreover, the positive and negative samples have to include samples acquired under different environment conditions (e.g. sunlight from different angles, rain, or fog). It is also proposed to include parts of traffic participants into the set of negative samples to reduce the amount of nested detections as well as the shadows caused by traffic participants.

An alternative method to reduce the number of false positives is to include additional sensors into the system to validate, classify, and track multiple traffic participants in the environment. The use of additional traffic participants and object tracking is proposed in the following.

Additional Sensors

The current system mainly relies on two sensors: a video camera and a single-beam laser scanner. Including both radar information and a 3-D laser scanner in the system provides the system with additional information. Possible 3-D laser scanners are a multi-beam

laser scanners or a rotatable single-beam laser scanner. Our investigations show that a system to generate efficient scanlines for rotatable laser scanners as proposed in section 5.3 provides a feasible extension to acquire sparse 3-D range data. A second option is to use the radar information already available in the multi-sensor system. While dismissed for the presented system in section 3.5.1, radar sensors as well as a 3-D laser scanner allow tracking of traffic participants, which is a third aspect of future work described in the following.

Introduction of Object Tracking

The proposed system operates on a frame-to-frame basis, no tracking of traffic participants is implemented. This is due to the additional complexity an object tracking and track fusion system introduces in a system. However, the tracking of traffic participant tracking method is beneficial to several core modules in our system.

First, tracking detected objects increases the robustness of both traffic participant localisation and traffic participant classification. The trajectory and identity of an object can be maintained over multiple cycles, even if the traffic participant is not detected and classified in every cycle. This in turn increases the effectiveness of our resource allocation system, as a tracked traffic participant does not require to be continually classified, enabling the system to allocate its sensor resources and computational resource on traffic participants with a higher uncertainty.

Second, the estimation of TTC using a traffic participant tracking system is preferable to using the range profile differentiation method used in this thesis. Knowledge about the traffic participant's trajectory and type allows the use of specific motion models for different traffic participant types. An adequate motion model in turn increases the robustness of both traffic participant tracking and determination of TTC.

Third, known traffic participant trajectories allow to predict occlusions, which presents the fourth aspect of future work.

Occlusion Detection

Occluded traffic participants cannot be observed in their entirety, therefore classification becomes harder or impossible in the case of total occlusion. In Magin *et al.*[216] two classes of algorithms able to determine whether an occlusion is present are discussed: object space

oriented, and image space oriented methods.

Of these, object space oriented methods have a high precision, as a ray tracing algorithm checks the borders of all known objects for intersections towards the sensor's focus with all other objects. This can either be performed on a 2-D plan view map or in 3-D space, depending on the sensor. Image space oriented methods often resolve occlusion problems using a z-buffer approach, thus rejecting surface elements that exhibit a higher distance than existing surface elements at the same pixel location.

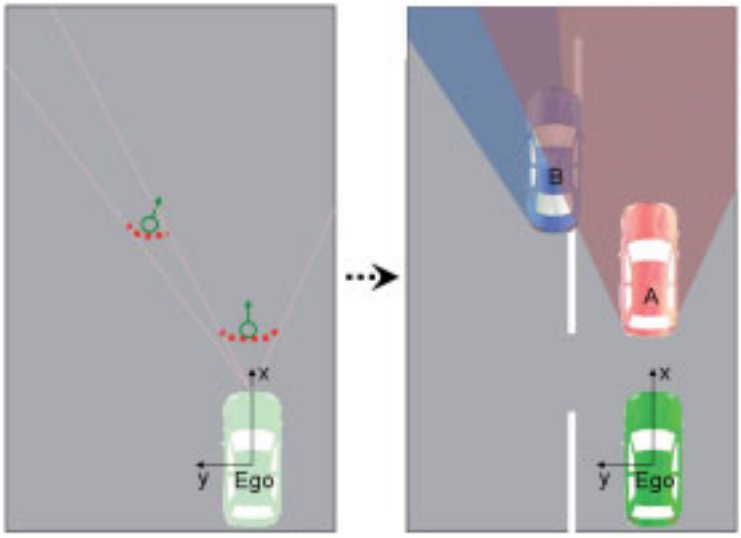


Figure 8.1: Plan view occlusion representation. Red dots represent laser readings, whereas green circles represent tracked radar targets with the arrow indicating the current velocity vector. Shaded areas are occluded by other surfaces, including parts of vehicle B by vehicle A.

Knowledge about occluded areas shown as shaded areas in Fig. 8.1 allow to determine whether an object's trajectory leads into an occluded area (immersing into occlusion), or out of an occluded area (emerging from occlusion). In the first case of an immersion into occlusion, an observation must be performed before the traffic participant is partially occluded or discarded entirely considering the actual utility of reducing the uncertainty about a traffic participant that is about to be occluded. In the second case, the observation of a traffic participant emerging from occlusion should ideally be postponed until the traffic participant completely emerges from occlusion. At this point however, the observation of a formerly unknown traffic participant is likely to provide a high utility.

The correlation between the utility of observing a traffic participant and its current or future occlusions suggests to include an occlusion measure in the list of objectives to be optimised, comparable with the use of TTC information for Ω_3.

Appendix A

Test Sequences

In this thesis a set of test sequences[1] is used to evaluate the proposed methods. In the following, five sequences given in Tab. A.1.

		Sensors		
Sequence	Fig.	Video	PMD	Laser
Tor	A.1	(x)	x	(x)
Pmd	A.2	(x)	x	x
Trc	A.3	x	(x)	
Urb	A.4	x	(x)	x
Mwy	A.5	x	(x)	x

Table A.1: Test sequences used for evaluation. Parentheses indicate that sensor data is available, but not used for evaluation.

[1] The video data of all test sequences is available online: http://www.matzka.net/vision/html/resources.html

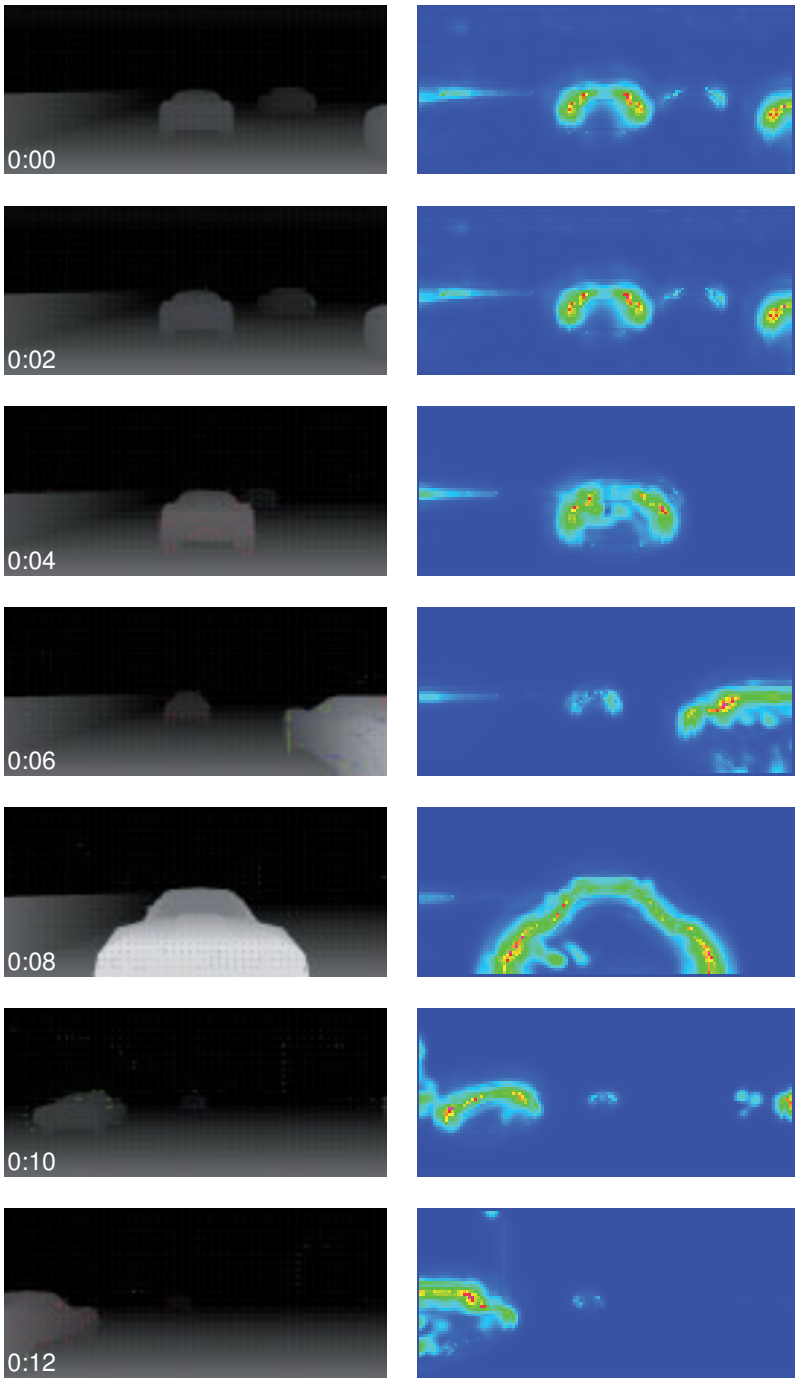


Figure A.1: Torcs sequence (Tor) acquired over 10 seconds with a frame rate of 15fps. The depth buffer of the Open Source racing game Torcs with estimated motion vectors and the respective saliency maps are shown.

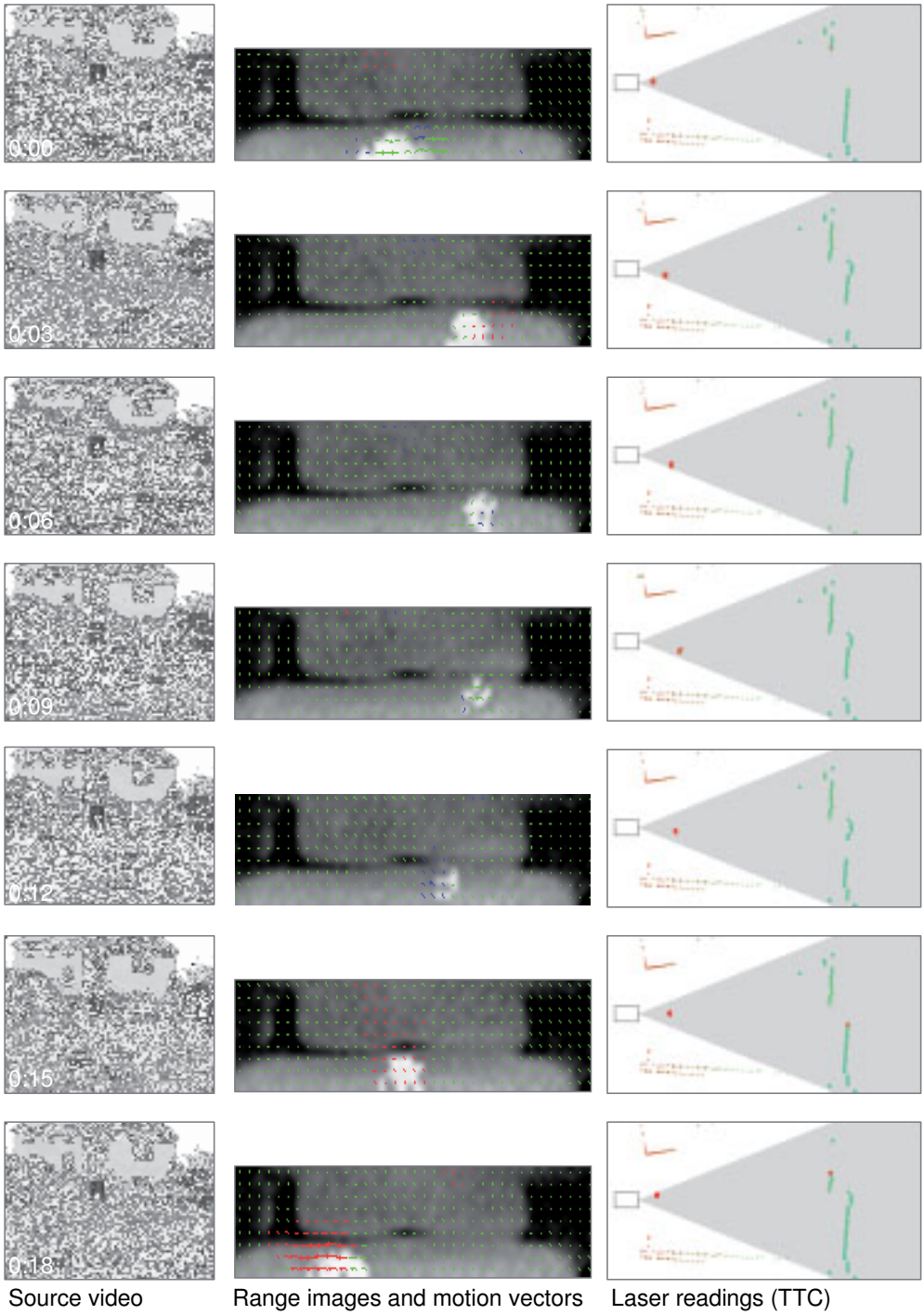


Figure A.2: PMD sensor sequence (PMD) acquired over 18 seconds using a video camera, a PMD camera, and a laser scanner. The estimated 3-D motion vectors are shown for the range map.

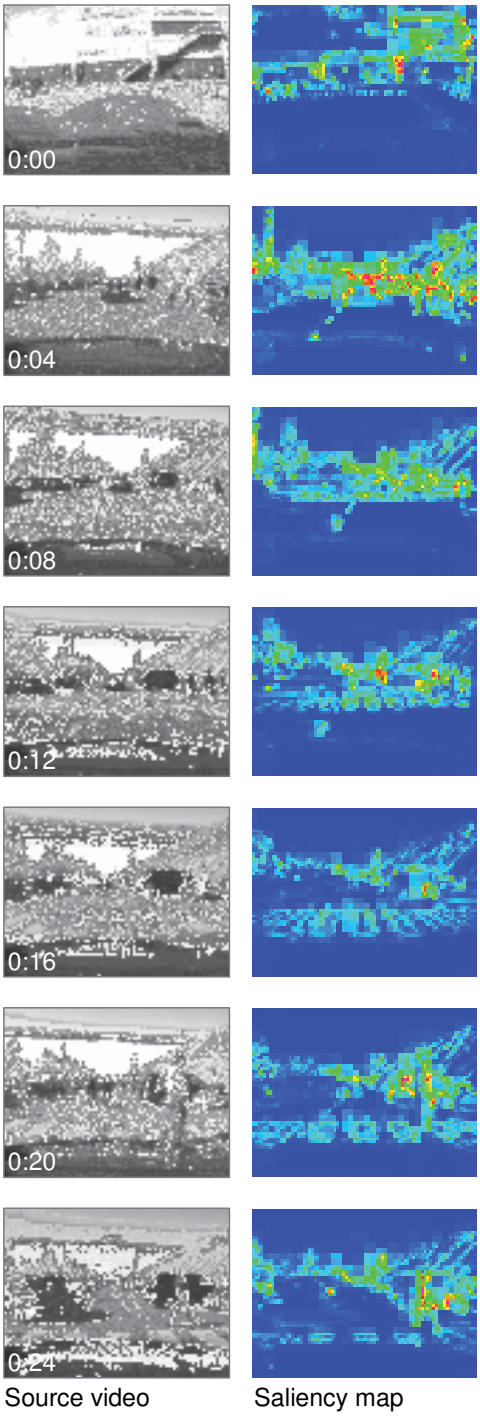


Figure A.3: Traffic calmed road sequence (TRC) recorded over 30s with a video frame rate of 25fps. No range information is available for this sequence.

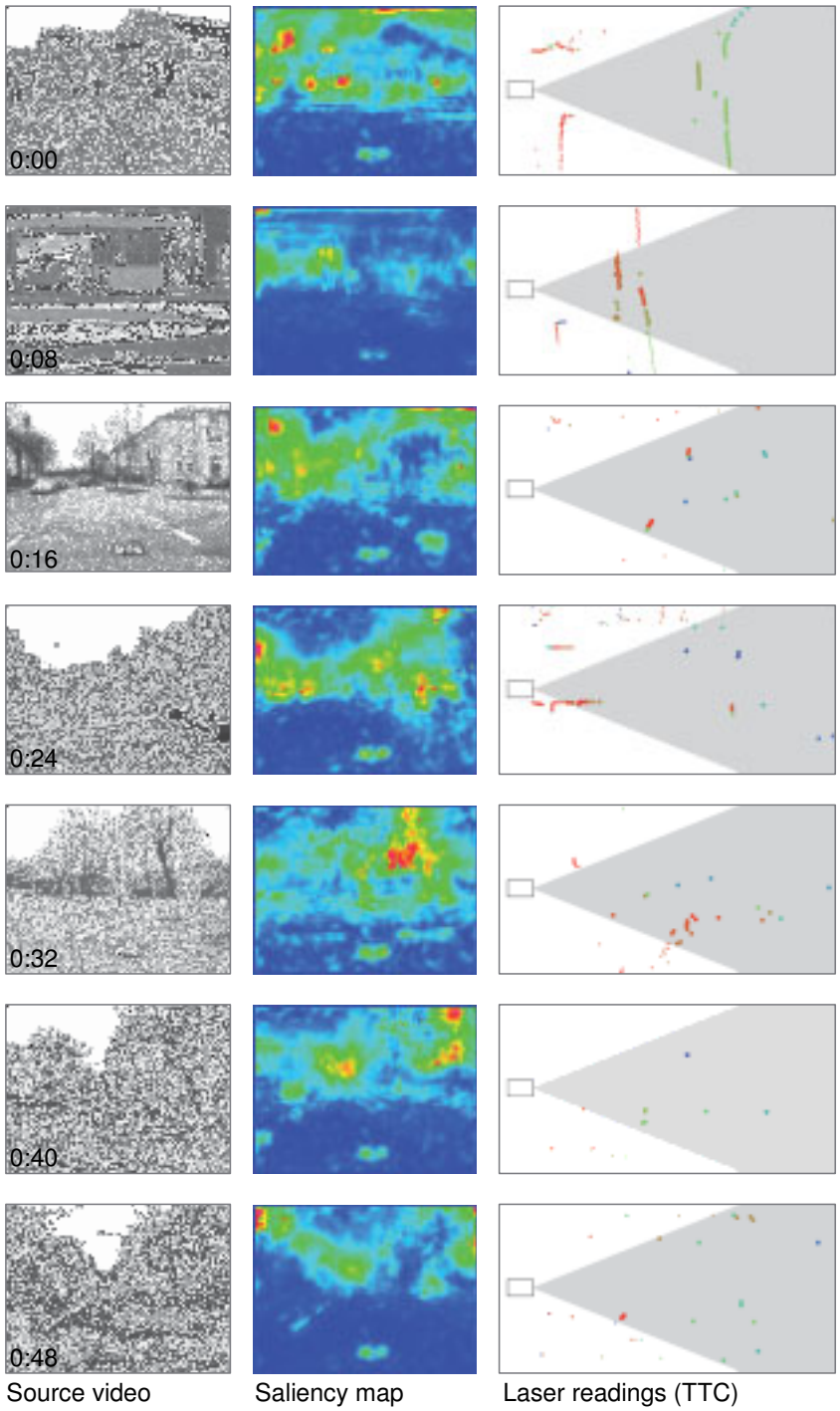


Figure A.4: Urban road sequence (URB) recorded over 50s with a video framerate of 25fps and a laser scanner rate of 75Hz. The colour of the laser scanner readings indicates the estimated time-to-collision ranging from <0.5s (red) to >5s (blue).

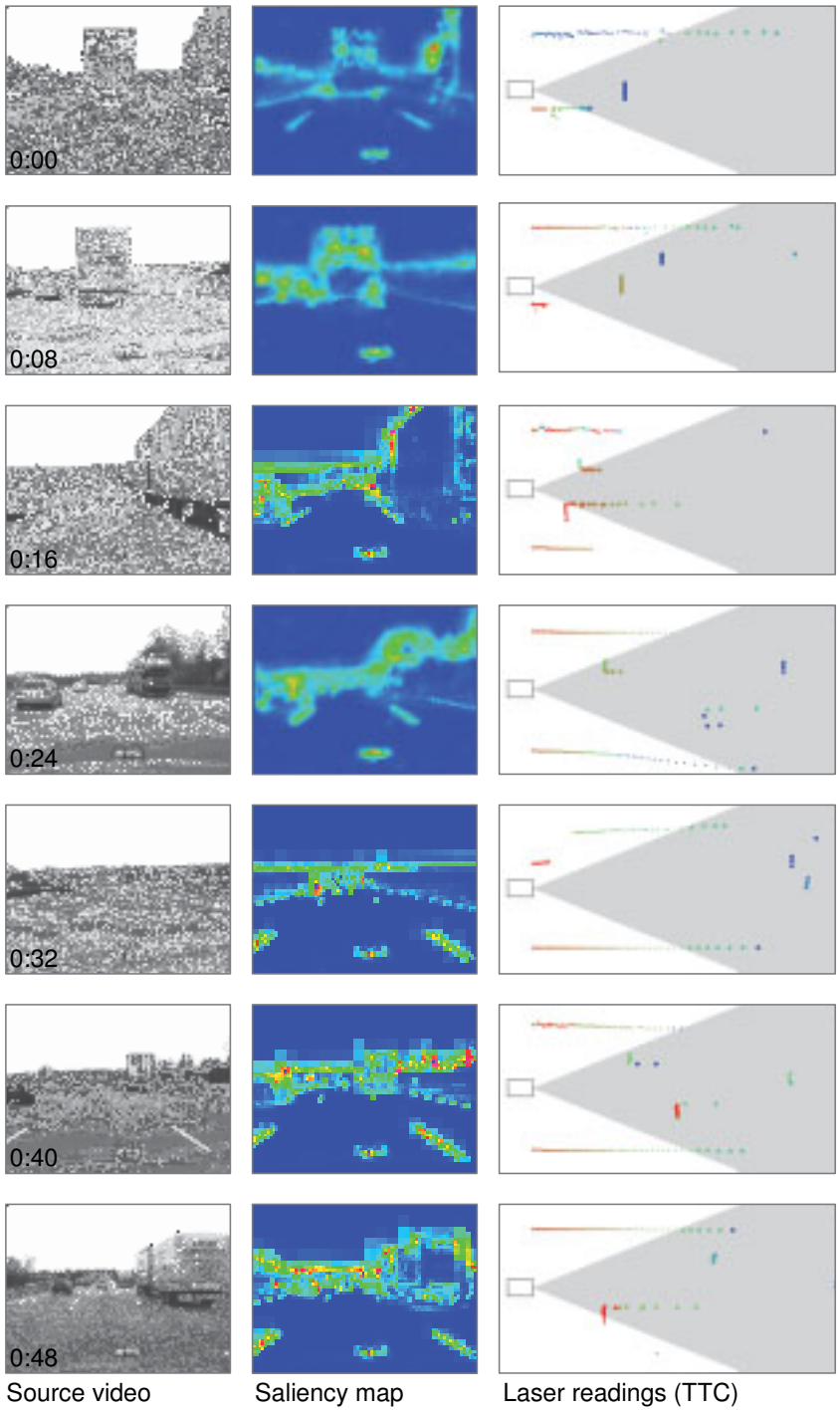


Figure A.5: Motorway sequence (MWY) recorded over 50s with a video framerate of 25fps and a laser scanner rate of 75Hz. The colour of the laser scanner readings indicates the estimated time-to-collision ranging from <0.5s (red) to >5s (blue).

Appendix B

Resulting Allocations

In order to examine the resulting allocations for three test sequences (TRC, URB, and MWY), a subset of six short frame sequences from the complete sequence is selected to represent both operation with auxiliary traffic participant detection results and operation under the adverse influence of false positive detections. For this, three sequential video frames are shown on every page.

For every frame, the source frame with detected human traffic participants (orange) and detected vehicles (green) is given in the upper left corner. Also in the source frame, the numbered candidate regions can be seen. In the lower left corner, the global traffic participant probability obtained from the current frame is shown for all traffic participant types $\mathcal{TP}_{n=1,..,5}$. In the middle column, the candidate regions selected for observation (drawn red in the source frame) for two simulated focused sensors are shown. The combined utility $\mathcal{U}^u(\mathcal{R}_n)$ of every candidate region $\mathcal{R}_n$, used to determine the regions to be observed, is shown in the right column. Below this list of combined utilities, the static classifier queue determined by the computational resource allocation is given. For this, classifier priority $\mathcal{P}$, sensor $\mathcal{S}$, region $\mathcal{R}$, classifier type and scaling $\mathcal{C}_{n,m}$, and mean classifier execution time are given. This structure is also illustrated in Fig. B.1.

While the image sequences are largely self-explaining, a short description and qualitative evaluation of the allocation is given before the respective sequences. After each description, the quantitative performance of the classifiers is compared to the performance of the detectors. For this, the number of true positives $N_{(C,\mathcal{TP})}$ using the allocated classifiers as well as the detectors (in parentheses) inside the selected candidate regions is

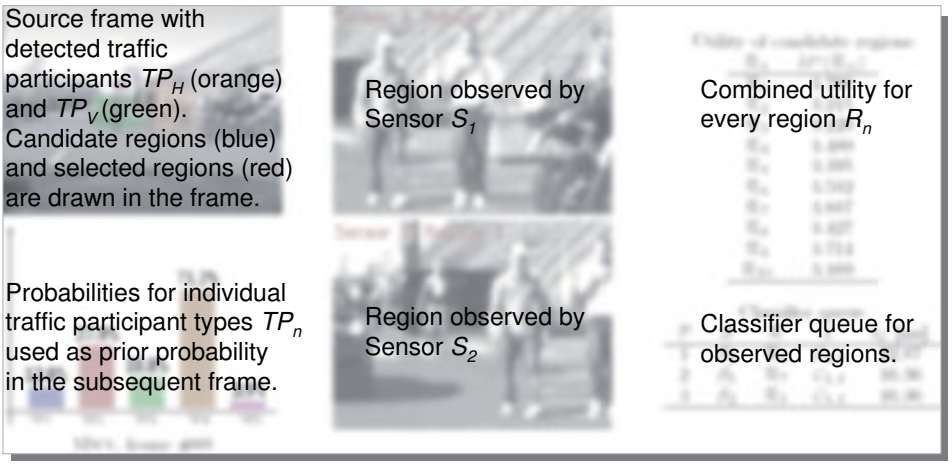


Figure B.1: Example structure of a single frame representation.

given. The same is done for false positives. Ideally, the number of true positives increases with classification (e.g. $N_{(C,\mathcal{TP})} = 4(2)$) and the number of false positives decreases (e.g. $N_{(C,\neg\mathcal{TP})} = 1(3)$).

B.1 Allocation for Traffic Calmed Sequence (Trc)

Examples with auxiliary Detection Results

- Frames #60–62 (p. 242) of the TRC sequence show two continually detected human traffic participants and one vehicle. This ensures a good allocation of sensor resources and an adequate classifier queue.

  $\Rightarrow \quad N_{(C,\mathcal{IP})} = 7(7),\ N_{(C,\neg\mathcal{IP})} = 2(1)$

- Frames #210–212 (p. 244) of the TRC sequence both the bicyclist and the women on the right side of the frame are correctly detected, allocating the sensor resources and classifiers accordingly.

  $\Rightarrow \quad N_{(C,\mathcal{IP})} = 7(6),\ N_{(C,\neg\mathcal{IP})} = 1(0)$

- Frames #450–452 (p. 247) of the TRC sequence show a majority of correct detections. Both the bicyclist and the pedestrian on the left side of the frame are focused.

  $\Rightarrow \quad N_{(C,\mathcal{IP})} = 3(1),\ N_{(C,\neg\mathcal{IP})} = 2(2)$

Examples with adverse Detection Results

- Frames #110–112 (p. 243) of the TRC sequence exhibit six false positive vehicle detections and fail to detect the near pedestrian in two frames. Accordingly, the classifier priorities are shifted to car classifiers.

  $\Rightarrow \quad N_{(C,\mathcal{IP})} = 6(4),\ N_{(C,\neg\mathcal{IP})} = 2(4)$

- Frames #240–242 (p. 245) of the TRC sequence contains a misleading detected car on the right side of the frame, which is focused and processed using a car classifier.

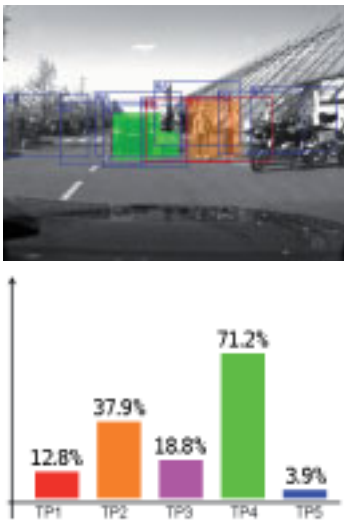
  $\Rightarrow \quad N_{(C,\mathcal{IP})} = 6(8),\ N_{(C,\neg\mathcal{IP})} = 1(3)$

- Frames #360–362 (p. 246) of the TRC sequence show both a misleading salient region $\mathcal{R}_6$ and a false positive vehicle detection on the right side, leading to an inadequate priority for car classifiers.

  $\Rightarrow \quad N_{(C,\mathcal{IP})} = 4(1),\ N_{(C,\neg\mathcal{IP})} = 2(2)$

Detection	True positives		False positives	
results are	Classification	Detection	Classification	Detection
Auxiliary	17	14	5	3
Adverse	16	13	5	9
Total	33	27	10	12

Table B.1: True positives and false positives for detectors and allocated classifiers for the selected frames in TRC.



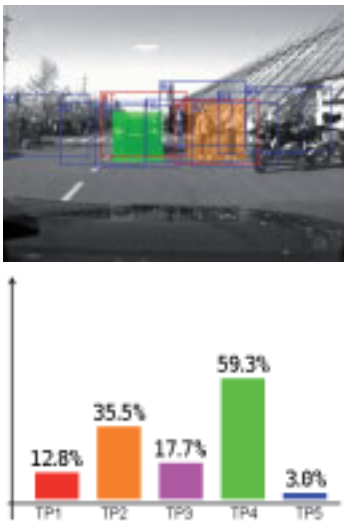
TRC, frame #60

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.798
$\mathcal{R}_2$	3.215
$\mathcal{R}_3$	3.220
$\mathcal{R}_4$	3.480
$\mathcal{R}_5$	3.395
$\mathcal{R}_6$	3.502
$\mathcal{R}_7$	3.887
$\mathcal{R}_8$	3.427
$\mathcal{R}_9$	3.714
$\mathcal{R}_{10}$	3.389

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_1$	$\mathcal{C}_{1,1}$	16.36



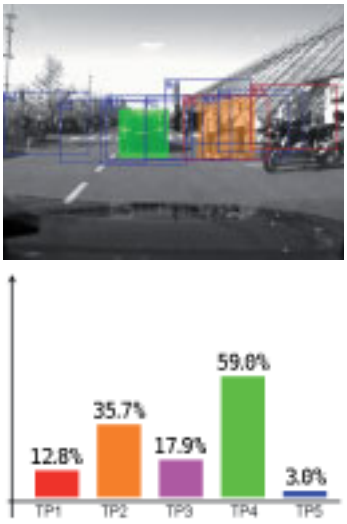
TRC, frame #61

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.183
$\mathcal{R}_2$	3.364
$\mathcal{R}_3$	3.313
$\mathcal{R}_4$	3.476
$\mathcal{R}_5$	3.564
$\mathcal{R}_6$	3.424
$\mathcal{R}_7$	3.570
$\mathcal{R}_8$	3.499
$\mathcal{R}_9$	3.837
$\mathcal{R}_{10}$	3.496

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87



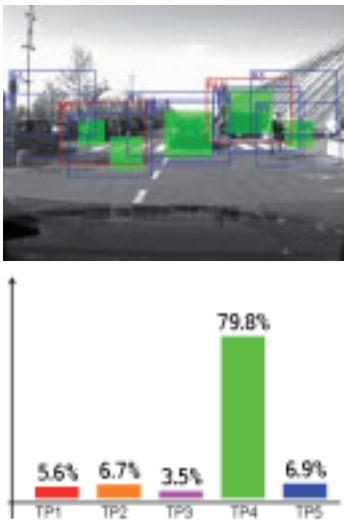
TRC, frame #62

Utility of candidate regions

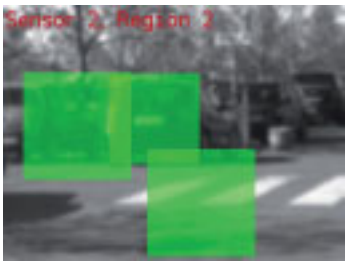
$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.545
$\mathcal{R}_2$	3.233
$\mathcal{R}_3$	3.336
$\mathcal{R}_4$	3.500
$\mathcal{R}_5$	3.629
$\mathcal{R}_6$	3.479
$\mathcal{R}_7$	3.416
$\mathcal{R}_8$	3.595
$\mathcal{R}_9$	3.889
$\mathcal{R}_{10}$	3.259

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_5$	$\mathcal{C}_{4,1}$	22.87



TRC, frame #110

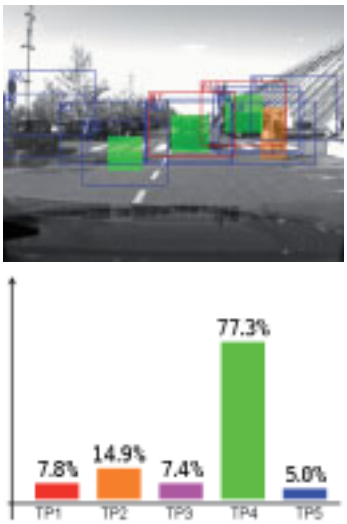


Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	4.031
$\mathcal{R}_2$	4.045
$\mathcal{R}_3$	3.527
$\mathcal{R}_4$	3.680
$\mathcal{R}_5$	3.872
$\mathcal{R}_6$	3.731
$\mathcal{R}_7$	3.892
$\mathcal{R}_8$	3.797
$\mathcal{R}_9$	3.945
$\mathcal{R}_{10}$	4.019
$\mathcal{R}_{11}$	3.994
$\mathcal{R}_{12}$	3.987

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{11}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_2$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{11}$	$\mathcal{C}_{1,1}$	16.36



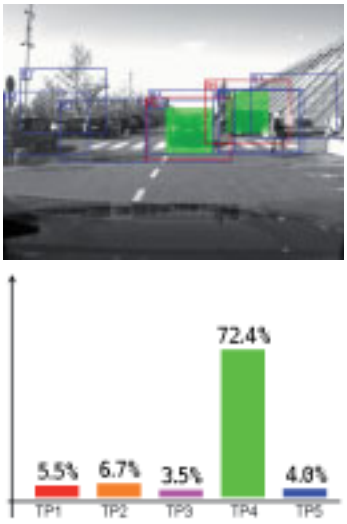
TRC, frame #111

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.789
$\mathcal{R}_2$	3.399
$\mathcal{R}_3$	3.661
$\mathcal{R}_4$	3.571
$\mathcal{R}_5$	3.747
$\mathcal{R}_6$	3.632
$\mathcal{R}_7$	3.844
$\mathcal{R}_8$	3.717
$\mathcal{R}_9$	3.771
$\mathcal{R}_{10}$	3.634
$\mathcal{R}_{11}$	3.757
$\mathcal{R}_{12}$	3.842

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_{12}$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36



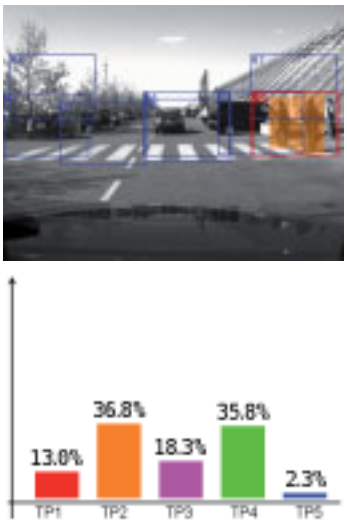
TRC, frame #112

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	4.025
$\mathcal{R}_2$	3.612
$\mathcal{R}_3$	3.948
$\mathcal{R}_4$	3.793
$\mathcal{R}_5$	3.892
$\mathcal{R}_6$	3.786
$\mathcal{R}_7$	3.778
$\mathcal{R}_8$	3.961
$\mathcal{R}_9$	3.993
$\mathcal{R}_{10}$	3.983

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{1,1}$	16.36



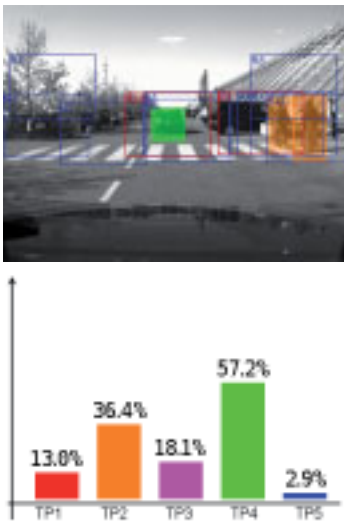
TRC, frame #210

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.508
$\mathcal{R}_2$	3.535
$\mathcal{R}_3$	3.706
$\mathcal{R}_4$	3.533
$\mathcal{R}_5$	3.967
$\mathcal{R}_6$	3.559
$\mathcal{R}_7$	3.604
$\mathcal{R}_8$	3.456
$\mathcal{R}_9$	3.956

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_5$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_1$	$\mathcal{R}_5$	$\mathcal{C}_{4,1}$	22.87



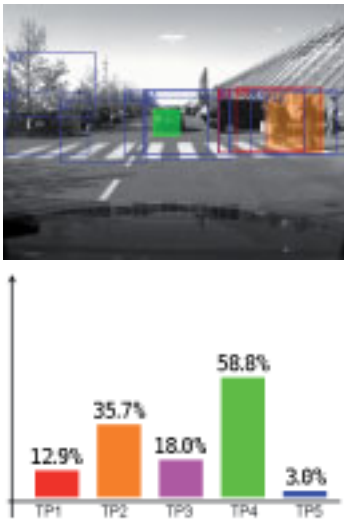
TRC, frame #211

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.532
$\mathcal{R}_2$	3.477
$\mathcal{R}_3$	3.654
$\mathcal{R}_4$	3.550
$\mathcal{R}_5$	3.563
$\mathcal{R}_6$	3.590
$\mathcal{R}_7$	3.494
$\mathcal{R}_8$	3.599
$\mathcal{R}_9$	3.605
$\mathcal{R}_{10}$	3.557
$\mathcal{R}_{11}$	2.765

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87



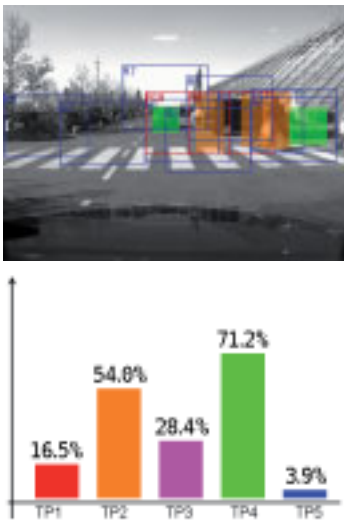
TRC, frame #212

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.829
$\mathcal{R}_2$	3.670
$\mathcal{R}_3$	4.026
$\mathcal{R}_4$	3.737
$\mathcal{R}_5$	3.969
$\mathcal{R}_6$	3.889
$\mathcal{R}_7$	3.696
$\mathcal{R}_8$	4.026
$\mathcal{R}_9$	4.036
$\mathcal{R}_{10}$	3.827
$\mathcal{R}_{11}$	3.884

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87



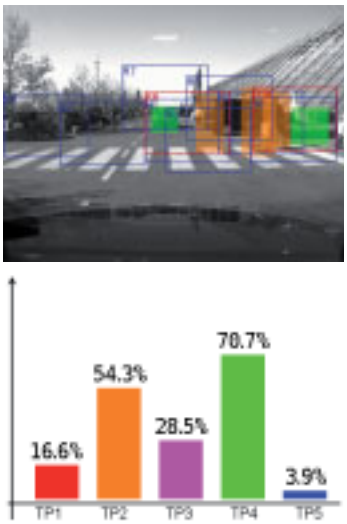
TRC, frame #240

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.617
$\mathcal{R}_2$	3.620
$\mathcal{R}_3$	3.918
$\mathcal{R}_4$	3.502
$\mathcal{R}_5$	3.511
$\mathcal{R}_6$	3.514
$\mathcal{R}_7$	3.728
$\mathcal{R}_8$	3.276
$\mathcal{R}_9$	4.022
$\mathcal{R}_{10}$	4.025
$\mathcal{R}_{11}$	3.914

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36



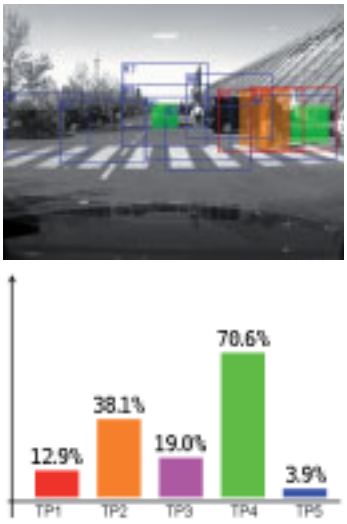
TRC, frame #241

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.609
$\mathcal{R}_2$	3.709
$\mathcal{R}_3$	3.705
$\mathcal{R}_4$	3.545
$\mathcal{R}_5$	3.538
$\mathcal{R}_6$	3.596
$\mathcal{R}_7$	3.782
$\mathcal{R}_8$	3.277
$\mathcal{R}_9$	4.068
$\mathcal{R}_{10}$	4.076

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36



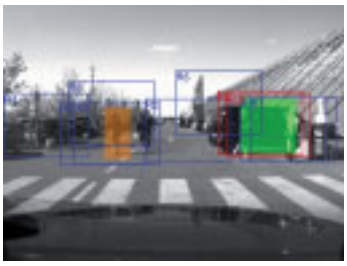
TRC, frame #242

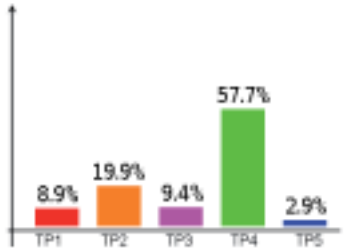
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.428
$\mathcal{R}_2$	3.710
$\mathcal{R}_3$	3.827
$\mathcal{R}_4$	3.517
$\mathcal{R}_5$	3.423
$\mathcal{R}_6$	3.495
$\mathcal{R}_7$	3.689
$\mathcal{R}_8$	3.172
$\mathcal{R}_9$	3.666
$\mathcal{R}_{10}$	3.895

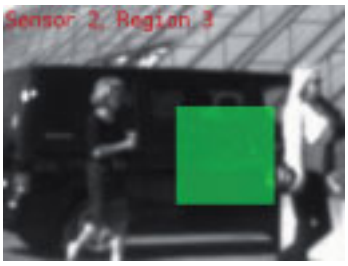
Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36





TRC, frame #360

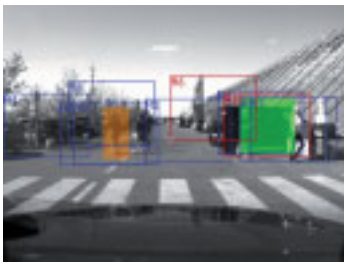


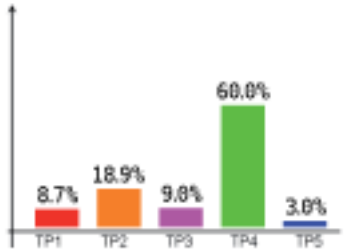
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.401
$\mathcal{R}_2$	3.553
$\mathcal{R}_3$	3.715
$\mathcal{R}_4$	3.456
$\mathcal{R}_5$	3.713
$\mathcal{R}_6$	3.636
$\mathcal{R}_7$	3.526
$\mathcal{R}_8$	3.349
$\mathcal{R}_9$	3.555
$\mathcal{R}_{10}$	3.737

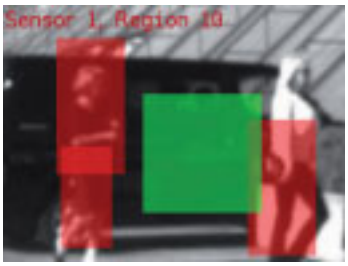
Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36





TRC, frame #361

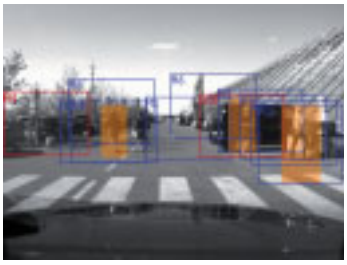


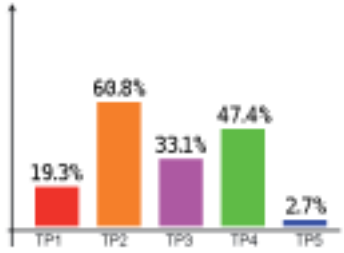
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.417
$\mathcal{R}_2$	3.612
$\mathcal{R}_3$	3.617
$\mathcal{R}_4$	3.496
$\mathcal{R}_5$	3.483
$\mathcal{R}_6$	3.621
$\mathcal{R}_7$	3.529
$\mathcal{R}_8$	3.376
$\mathcal{R}_9$	3.593
$\mathcal{R}_{10}$	3.659

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36





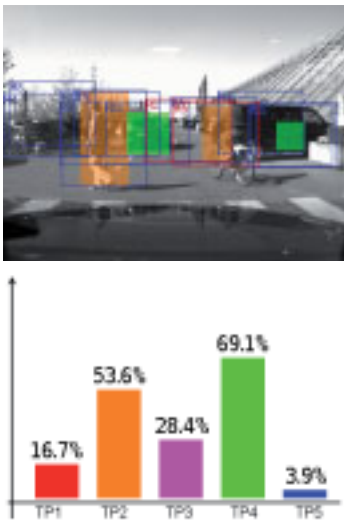
TRC, frame #362

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.181
$\mathcal{R}_2$	3.400
$\mathcal{R}_3$	3.248
$\mathcal{R}_4$	3.418
$\mathcal{R}_5$	3.353
$\mathcal{R}_6$	3.266
$\mathcal{R}_7$	3.411
$\mathcal{R}_8$	3.347
$\mathcal{R}_9$	3.395
$\mathcal{R}_{10}$	3.399
$\mathcal{R}_{11}$	3.108

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36



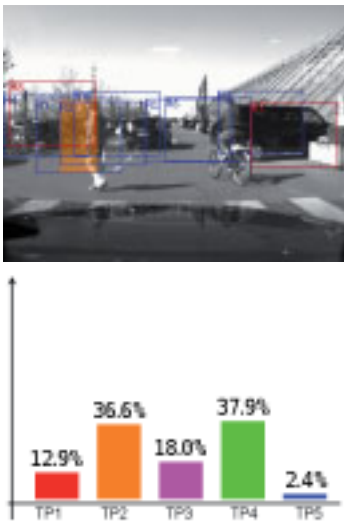
TRC, frame #450

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.621
$\mathcal{R}_2$	3.495
$\mathcal{R}_3$	3.269
$\mathcal{R}_4$	3.491
$\mathcal{R}_5$	3.474
$\mathcal{R}_6$	3.259
$\mathcal{R}_7$	3.574
$\mathcal{R}_8$	3.582
$\mathcal{R}_9$	3.601
$\mathcal{R}_{10}$	3.458
$\mathcal{R}_{11}$	3.437
$\mathcal{R}_{12}$	3.060

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87



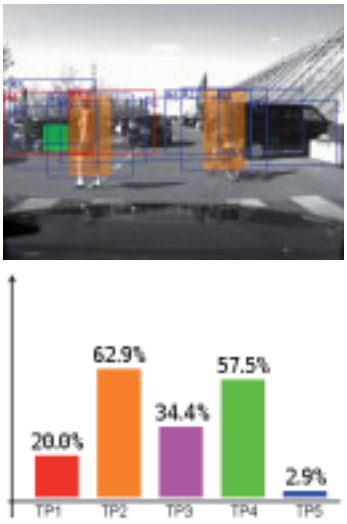
TRC, frame #451

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.121
$\mathcal{R}_2$	3.338
$\mathcal{R}_3$	3.190
$\mathcal{R}_4$	3.257
$\mathcal{R}_5$	3.356
$\mathcal{R}_6$	3.125
$\mathcal{R}_7$	3.423
$\mathcal{R}_8$	3.268
$\mathcal{R}_9$	2.885

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_8$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_8$	$\mathcal{C}_{1,1}$	16.36



TRC, frame #452

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.298
$\mathcal{R}_2$	3.423
$\mathcal{R}_3$	3.229
$\mathcal{R}_4$	3.398
$\mathcal{R}_5$	3.501
$\mathcal{R}_6$	3.319
$\mathcal{R}_7$	3.413
$\mathcal{R}_8$	3.275
$\mathcal{R}_9$	3.328
$\mathcal{R}_{10}$	2.868
$\mathcal{R}_{11}$	2.894

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_5$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{1,1}$	16.36

B.2 Allocation for Urban Sequence (Urb)

Examples with auxiliary Detection Results

- Frames #250–252 (p. 250) of the URB sequence show a detected bicyclist, yet also a false positive vehicle on the right side.

  $\Rightarrow \quad N_{(C,\mathcal{I}\mathcal{P})} = 2(3),\ N_{(C,\neg\mathcal{I}\mathcal{P})} = 5(2)$

- Frames #580–582 (p. 252) of the URB sequence contains no detections, using saliency and TTC information to select the focused regions.

  $\Rightarrow \quad N_{(C,\mathcal{I}\mathcal{P})} = 0(0),\ N_{(C,\neg\mathcal{I}\mathcal{P})} = 5(0)$

- Frames #700–702 (p. 254) of the URB sequence exhibits few detections, but succeeds in selecting suitable regions and classifiers.

  $\Rightarrow \quad N_{(C,\mathcal{I}\mathcal{P})} = 3(1),\ N_{(C,\neg\mathcal{I}\mathcal{P})} = 3(1)$

Examples with adverse Detection Results

- Frames #90–92 (p. 249) of the URB sequence contain five false positive vehicle detections, leading to an inadequate priority for car classifiers.

  $\Rightarrow \quad N_{(C,\mathcal{I}\mathcal{P})} = 0(0),\ N_{(C,\neg\mathcal{I}\mathcal{P})} = 4(3)$

- Frames #300–302 (p. 251) of the URB sequence show the misleading effect of false positives on our resource allocation system.

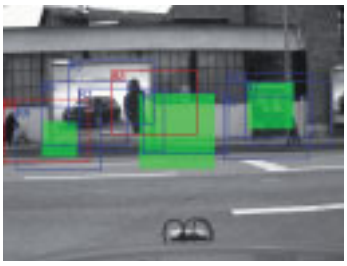
  $\Rightarrow \quad N_{(C,\mathcal{I}\mathcal{P})} = 1(0),\ N_{(C,\neg\mathcal{I}\mathcal{P})} = 6(3)$

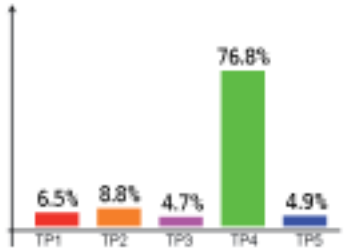
- Frames #640–642 (p. 253) of the URB sequence exhibit two false positive detections, yet the selected regions and classifiers are acceptable.

  $\Rightarrow \quad N_{(C,\mathcal{I}\mathcal{P})} = 2(0),\ N_{(C,\neg\mathcal{I}\mathcal{P})} = 3(0)$

Detection	True positives		False positives	
results are	Classification	Detection	Classification	Detection
Auxiliary	5	4	13	3
Adverse	3	0	13	6
Total	8	4	26	9

Table B.2: True positives and false positives for detectors and allocated classifiers for the selected frames in URB.





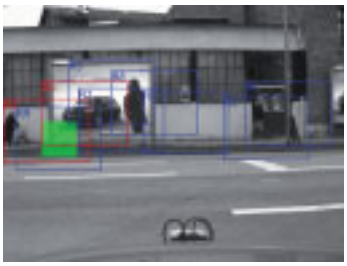
URB, frame #090

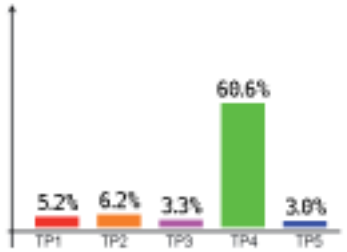
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.388
$\mathcal{R}_2$	3.546
$\mathcal{R}_3$	3.236
$\mathcal{R}_4$	3.637
$\mathcal{R}_5$	2.872
$\mathcal{R}_6$	3.664
$\mathcal{R}_7$	3.571
$\mathcal{R}_8$	3.772
$\mathcal{R}_9$	3.558
$\mathcal{R}_{10}$	3.763

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_8$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{1,1}$	16.36





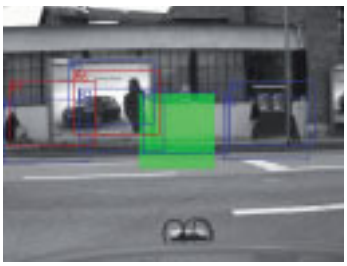
URB, frame #091

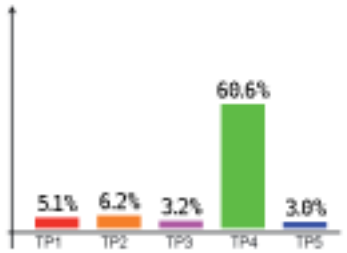
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.212
$\mathcal{R}_2$	3.386
$\mathcal{R}_3$	3.390
$\mathcal{R}_4$	3.459
$\mathcal{R}_5$	2.953
$\mathcal{R}_6$	3.496
$\mathcal{R}_7$	3.388
$\mathcal{R}_8$	3.476
$\mathcal{R}_9$	3.371

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{1,1}$	16.36





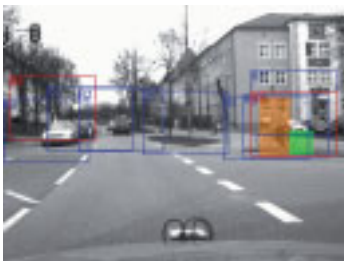
URB, frame #092

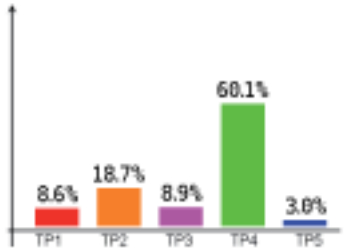
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.744
$\mathcal{R}_2$	3.803
$\mathcal{R}_3$	3.691
$\mathcal{R}_4$	3.647
$\mathcal{R}_5$	3.181
$\mathcal{R}_6$	3.973
$\mathcal{R}_7$	3.837
$\mathcal{R}_8$	3.793

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{1,1}$	16.36





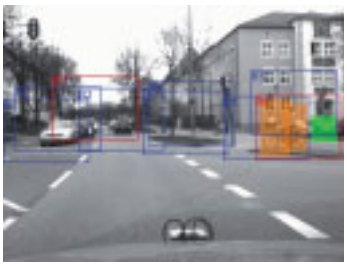
URB, frame #250

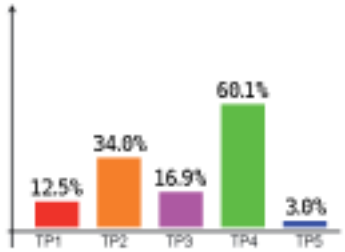
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.167
$\mathcal{R}_2$	3.187
$\mathcal{R}_3$	3.183
$\mathcal{R}_4$	3.596
$\mathcal{R}_5$	3.528
$\mathcal{R}_6$	3.666
$\mathcal{R}_7$	3.628
$\mathcal{R}_8$	3.726
$\mathcal{R}_9$	3.642

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_8$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{1,1}$	16.36





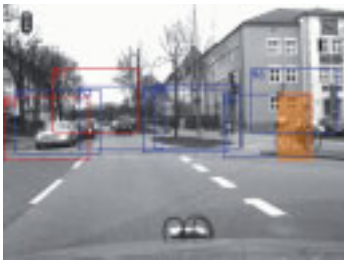
URB, frame #251

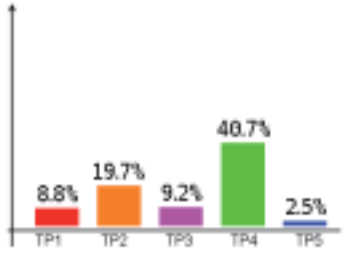
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	2.748
$\mathcal{R}_2$	3.062
$\mathcal{R}_3$	3.347
$\mathcal{R}_4$	3.270
$\mathcal{R}_5$	2.641
$\mathcal{R}_6$	3.251
$\mathcal{R}_7$	3.499
$\mathcal{R}_8$	2.857
$\mathcal{R}_9$	3.584

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87





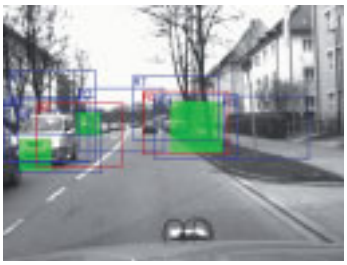
URB, frame #252

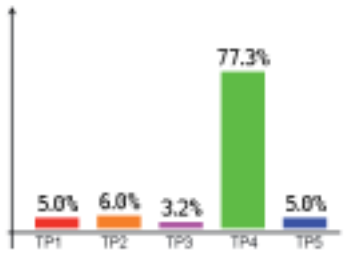
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.511
$\mathcal{R}_2$	3.265
$\mathcal{R}_3$	3.620
$\mathcal{R}_4$	4.105
$\mathcal{R}_5$	4.053
$\mathcal{R}_6$	3.793
$\mathcal{R}_7$	4.030
$\mathcal{R}_8$	3.476

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36





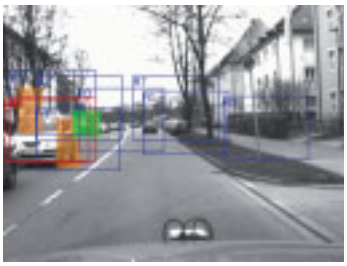
URB, frame #300

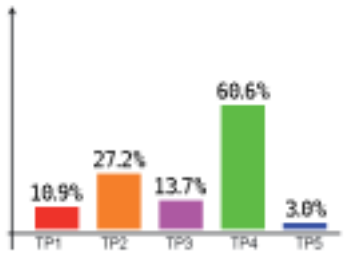
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.391
$\mathcal{R}_2$	2.956
$\mathcal{R}_3$	3.462
$\mathcal{R}_4$	3.371
$\mathcal{R}_5$	2.973
$\mathcal{R}_6$	3.327
$\mathcal{R}_7$	3.362
$\mathcal{R}_8$	3.499
$\mathcal{R}_9$	3.498
$\mathcal{R}_{10}$	3.354

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_8$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{1,1}$	16.36





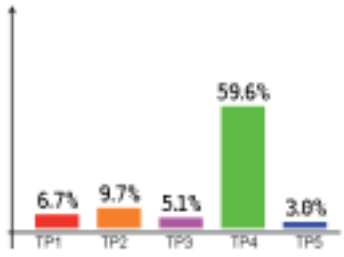
URB, frame #301

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	2.705
$\mathcal{R}_2$	2.307
$\mathcal{R}_3$	2.675
$\mathcal{R}_4$	3.473
$\mathcal{R}_5$	2.757
$\mathcal{R}_6$	3.422
$\mathcal{R}_7$	2.902
$\mathcal{R}_8$	2.917
$\mathcal{R}_9$	2.961

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{1,1}$	16.36



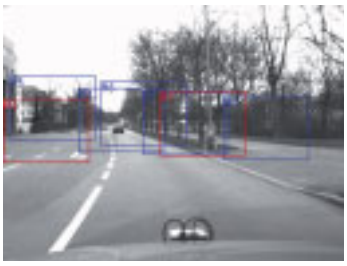
URB, frame #302

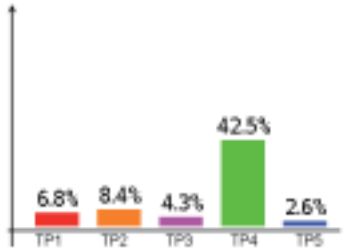
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	2.943
$\mathcal{R}_2$	2.593
$\mathcal{R}_3$	3.563
$\mathcal{R}_4$	3.222
$\mathcal{R}_5$	3.053
$\mathcal{R}_6$	3.140
$\mathcal{R}_7$	2.710
$\mathcal{R}_8$	3.059
$\mathcal{R}_9$	3.425

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36





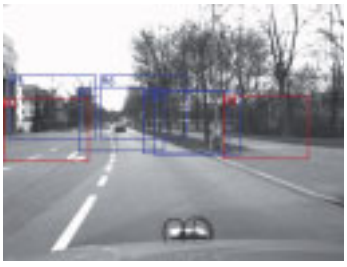
URB, frame #580

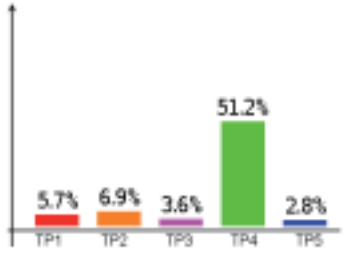
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.862
$\mathcal{R}_2$	3.202
$\mathcal{R}_3$	3.914
$\mathcal{R}_4$	3.967
$\mathcal{R}_5$	3.850
$\mathcal{R}_6$	3.123
$\mathcal{R}_7$	4.020

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36





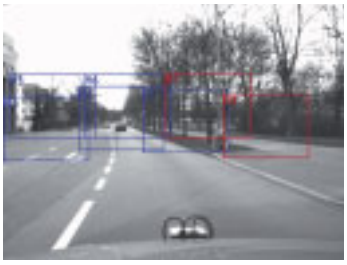
URB, frame #581

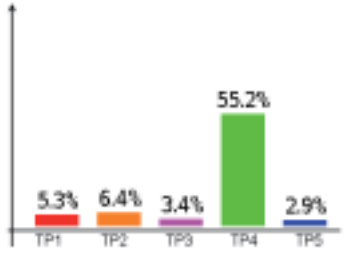
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.505
$\mathcal{R}_2$	3.336
$\mathcal{R}_3$	3.975
$\mathcal{R}_4$	3.809
$\mathcal{R}_5$	3.789
$\mathcal{R}_6$	3.538
$\mathcal{R}_7$	3.610

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36





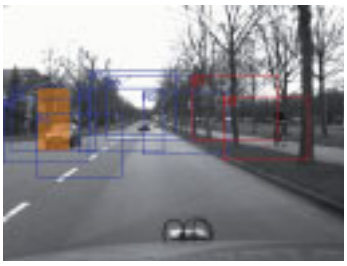
URB, frame #582

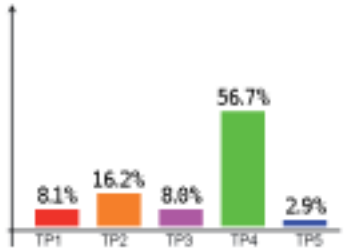
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.730
$\mathcal{R}_2$	3.061
$\mathcal{R}_3$	3.820
$\mathcal{R}_4$	3.396
$\mathcal{R}_5$	3.474
$\mathcal{R}_6$	3.110
$\mathcal{R}_7$	4.075

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36





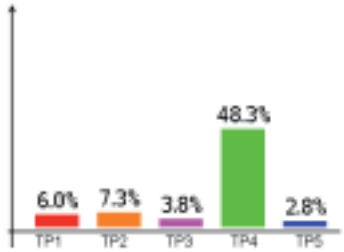
URB, frame #640

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.624
$\mathcal{R}_2$	3.180
$\mathcal{R}_3$	3.983
$\mathcal{R}_4$	3.166
$\mathcal{R}_5$	3.335
$\mathcal{R}_6$	2.963
$\mathcal{R}_7$	3.858
$\mathcal{R}_8$	3.479
$\mathcal{R}_9$	3.267

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36



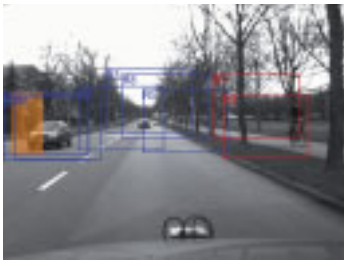
URB, frame #641

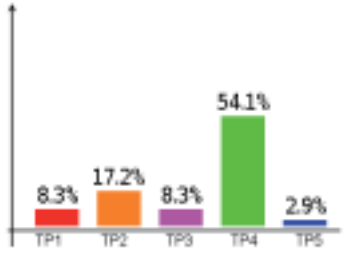
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.033
$\mathcal{R}_2$	3.244
$\mathcal{R}_3$	3.721
$\mathcal{R}_4$	3.544
$\mathcal{R}_5$	3.188
$\mathcal{R}_6$	3.459
$\mathcal{R}_7$	3.333
$\mathcal{R}_8$	3.099

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36





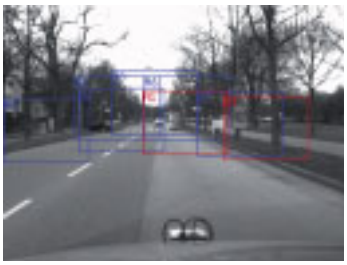
URB, frame #642

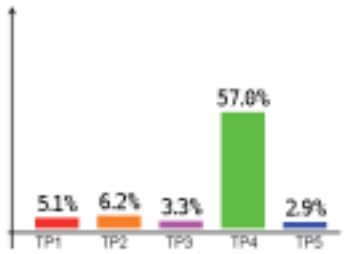
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.025
$\mathcal{R}_2$	3.303
$\mathcal{R}_3$	3.694
$\mathcal{R}_4$	3.264
$\mathcal{R}_5$	3.378
$\mathcal{R}_6$	3.255
$\mathcal{R}_7$	4.045
$\mathcal{R}_8$	3.241
$\mathcal{R}_9$	3.329

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36





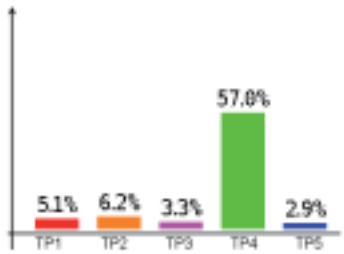
URB, frame #700

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.772
$\mathcal{R}_2$	3.598
$\mathcal{R}_3$	3.955
$\mathcal{R}_4$	3.510
$\mathcal{R}_5$	3.550
$\mathcal{R}_6$	3.768
$\mathcal{R}_7$	3.757
$\mathcal{R}_8$	3.701

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36



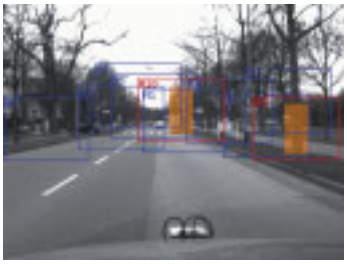
URB, frame #701

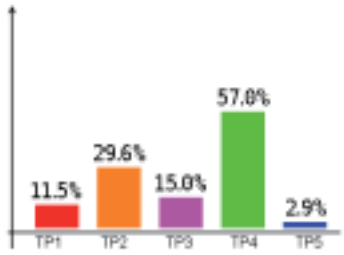
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.215
$\mathcal{R}_2$	3.439
$\mathcal{R}_3$	3.918
$\mathcal{R}_4$	3.585
$\mathcal{R}_5$	3.456
$\mathcal{R}_6$	3.350
$\mathcal{R}_7$	3.851
$\mathcal{R}_8$	3.509

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{1,1}$	16.36





URB, frame #702

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.136
$\mathcal{R}_2$	2.706
$\mathcal{R}_3$	3.027
$\mathcal{R}_4$	3.099
$\mathcal{R}_5$	3.015
$\mathcal{R}_6$	3.124
$\mathcal{R}_7$	3.444
$\mathcal{R}_8$	2.654
$\mathcal{R}_9$	3.646
$\mathcal{R}_{10}$	3.702

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36

B.3 Allocation for Motorway Sequence (Mwy)

Examples with auxiliary Detection Results

- Frames #90–92 (p. 256) of the Mwy sequence show that the false positive detection in one frame does not have an adverse impact on the resource allocation process.

  $\Rightarrow \quad N_{(C,\mathit{TP})} = 3(3),\ N_{(C,\neg\mathit{TP})} = 3(1)$

- Frames #490–492 (p. 259) of the Mwy sequence exhibit a generally good resource allocation in the presence of two correctly classified vehicles.

  $\Rightarrow \quad N_{(C,\mathit{TP})} = 4(3),\ N_{(C,\neg\mathit{TP})} = 1(0)$

- Frames #670–672 (p. 261) of the Mwy sequence contain four false positive vehicle detections but succeed in selecting adequate regions. The allocation of the lorry classifier $\mathcal{C}_{5,1}$ is disadvantageous.

  $\Rightarrow \quad N_{(C,\mathit{TP})} = 4(6),\ N_{(C,\neg\mathit{TP})} = 2(1)$

Examples with adverse Detection Results

- Frames #200–202 (p. 257) of the Mwy sequence show that the repeated false positive detection of human traffic participants lead to an inadequate classifier queue.

  $\Rightarrow \quad N_{(C,\mathit{TP})} = 1(0),\ N_{(C,\neg\mathit{TP})} = 1(7)$

- Frames #360–362 (p. 258) of the Mwy sequence contain two false positive human traffic participant detections, yet the selected regions and classifiers are acceptable.

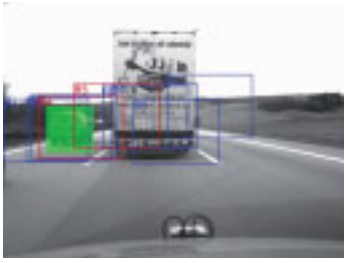
  $\Rightarrow \quad N_{(C,\mathit{TP})} = 4(2),\ N_{(C,\neg\mathit{TP})} = 6(3)$

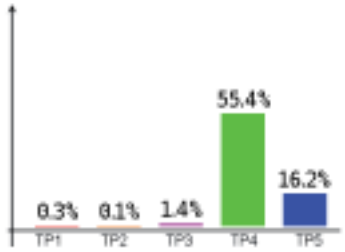
- Frames #650–652 (p. 260) of the Mwy sequence exhibit four false positive human traffic participant detections, resulting in an inadequate allocation of sensor resources and classifiers.

  $\Rightarrow \quad N_{(C,\mathit{TP})} = 3(3),\ N_{(C,\neg\mathit{TP})} = 2(6)$

Detection	True positives		False positives	
results are	Classification	Detection	Classification	Detection
Auxiliary	11	12	6	2
Adverse	8	5	9	16
Total	19	17	15	18

Table B.3: True positives and false positives for detectors and allocated classifiers for the selected frames in Mwy.





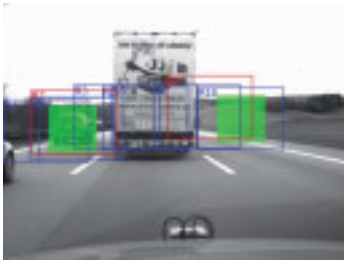
MWY, frame #90

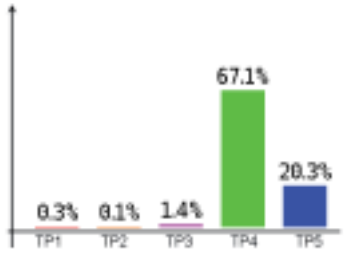
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.815
$\mathcal{R}_2$	3.997
$\mathcal{R}_3$	3.953
$\mathcal{R}_4$	3.611
$\mathcal{R}_5$	3.980
$\mathcal{R}_6$	3.797
$\mathcal{R}_7$	3.938
$\mathcal{R}_8$	3.795
$\mathcal{R}_9$	3.562

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_2$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_5$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_2$	$\mathcal{C}_{5,1}$	15.42





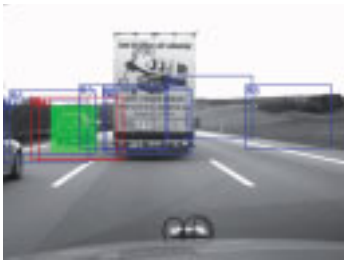
MWY, frame #91

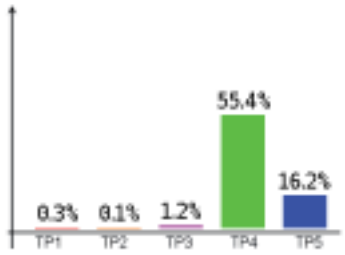
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.823
$\mathcal{R}_2$	3.961
$\mathcal{R}_3$	4.037
$\mathcal{R}_4$	3.614
$\mathcal{R}_5$	3.913
$\mathcal{R}_6$	3.816
$\mathcal{R}_7$	4.036
$\mathcal{R}_8$	3.913
$\mathcal{R}_9$	3.994
$\mathcal{R}_{10}$	3.213

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{5,1}$	15.42





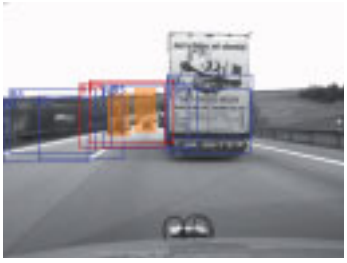
MWY, frame #92

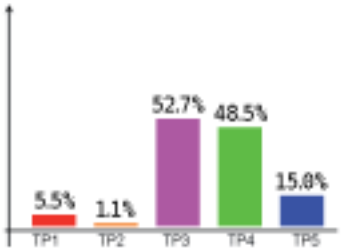
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.754
$\mathcal{R}_2$	3.902
$\mathcal{R}_3$	3.737
$\mathcal{R}_4$	3.644
$\mathcal{R}_5$	3.860
$\mathcal{R}_6$	3.680
$\mathcal{R}_7$	3.704
$\mathcal{R}_8$	3.373
$\mathcal{R}_9$	3.946

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_2$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{5,1}$	15.42





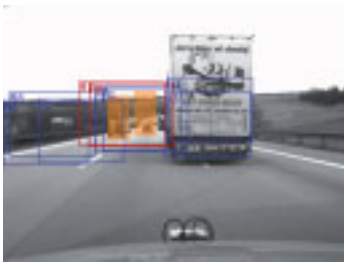
Mwy, frame #200

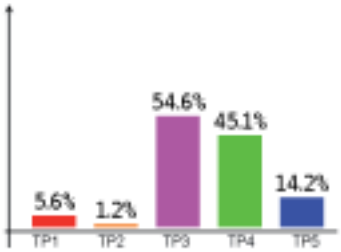
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.948
$\mathcal{R}_2$	2.679
$\mathcal{R}_3$	3.465
$\mathcal{R}_4$	3.681
$\mathcal{R}_5$	3.669
$\mathcal{R}_6$	3.383
$\mathcal{R}_7$	4.032
$\mathcal{R}_8$	3.719
$\mathcal{R}_9$	4.021

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87





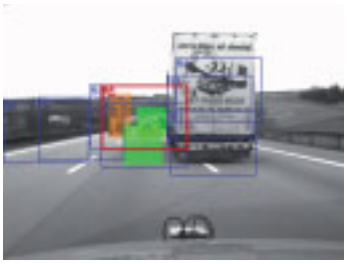
Mwy, frame #201

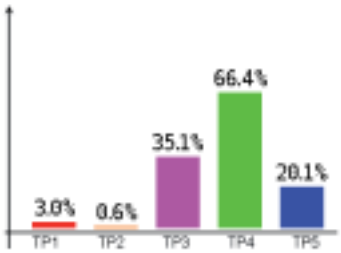
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.966
$\mathcal{R}_2$	2.738
$\mathcal{R}_3$	3.617
$\mathcal{R}_4$	3.012
$\mathcal{R}_5$	3.980
$\mathcal{R}_6$	3.405
$\mathcal{R}_7$	4.034
$\mathcal{R}_8$	3.032
$\mathcal{R}_9$	4.029

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_1$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87





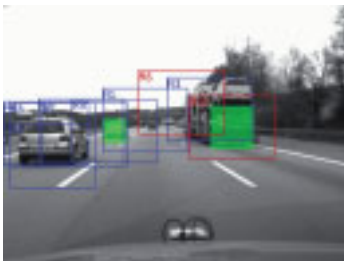
Mwy, frame #202

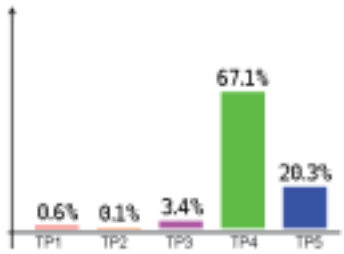
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.964
$\mathcal{R}_2$	2.720
$\mathcal{R}_3$	3.716
$\mathcal{R}_4$	3.042
$\mathcal{R}_5$	3.963
$\mathcal{R}_6$	3.031
$\mathcal{R}_7$	3.560
$\mathcal{R}_8$	3.804
$\mathcal{R}_9$	3.532

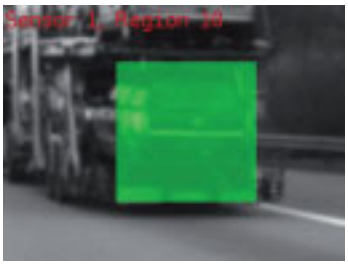
Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_1$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_5$	$\mathcal{C}_{1,1}$	16.36





Mwy, frame #360

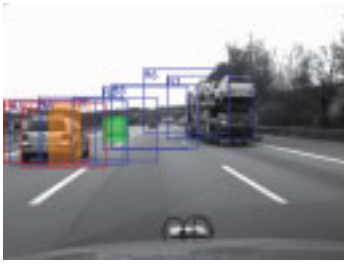


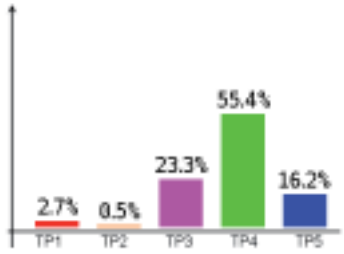
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.186
$\mathcal{R}_2$	3.239
$\mathcal{R}_3$	3.783
$\mathcal{R}_4$	3.331
$\mathcal{R}_5$	3.321
$\mathcal{R}_6$	3.887
$\mathcal{R}_7$	3.438
$\mathcal{R}_8$	3.166
$\mathcal{R}_9$	3.283
$\mathcal{R}_{10}$	3.670

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{5,1}$	15.42





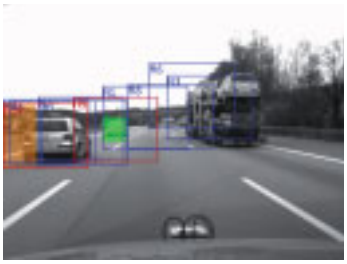
Mwy, frame #361

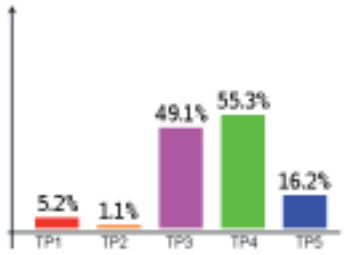
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	2.863
$\mathcal{R}_2$	3.551
$\mathcal{R}_3$	3.349
$\mathcal{R}_4$	3.629
$\mathcal{R}_5$	3.615
$\mathcal{R}_6$	3.347
$\mathcal{R}_7$	3.505
$\mathcal{R}_8$	3.007
$\mathcal{R}_9$	3.457
$\mathcal{R}_{10}$	3.957

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{5,1}$	15.42





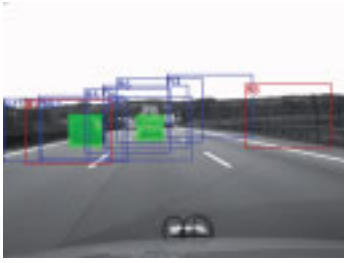
Mwy, frame #362

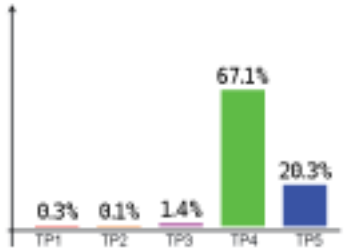
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	2.683
$\mathcal{R}_2$	3.310
$\mathcal{R}_3$	3.028
$\mathcal{R}_4$	3.326
$\mathcal{R}_5$	3.316
$\mathcal{R}_6$	3.078
$\mathcal{R}_7$	3.265
$\mathcal{R}_8$	2.913
$\mathcal{R}_9$	3.358
$\mathcal{R}_{10}$	3.957

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{1,1}$	16.36
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{10}$	$\mathcal{C}_{4,1}$	22.87





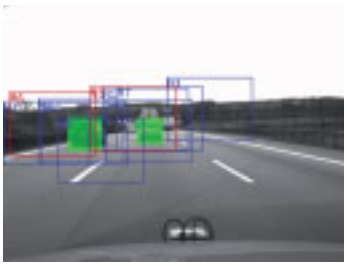
Mwy, frame #490

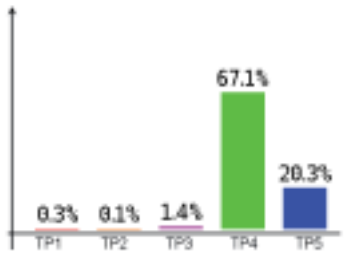
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.447
$\mathcal{R}_2$	3.518
$\mathcal{R}_3$	3.067
$\mathcal{R}_4$	3.462
$\mathcal{R}_5$	3.512
$\mathcal{R}_6$	3.444
$\mathcal{R}_7$	3.522
$\mathcal{R}_8$	3.554
$\mathcal{R}_9$	3.282
$\mathcal{R}_{10}$	3.506

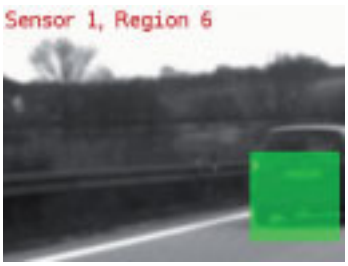
Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_8$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_7$	$\mathcal{C}_{5,1}$	15.42





Mwy, frame #491

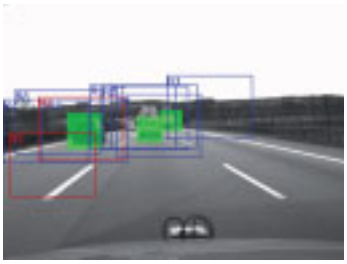


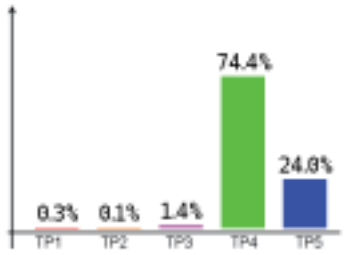
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.706
$\mathcal{R}_2$	3.694
$\mathcal{R}_3$	3.149
$\mathcal{R}_4$	3.712
$\mathcal{R}_5$	3.788
$\mathcal{R}_6$	3.790
$\mathcal{R}_7$	3.547
$\mathcal{R}_8$	3.493
$\mathcal{R}_9$	3.541
$\mathcal{R}_{10}$	3.676

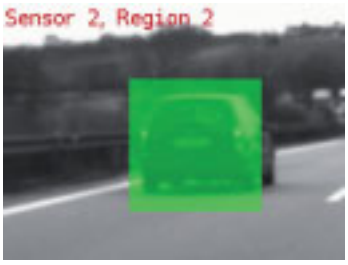
Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_5$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_5$	$\mathcal{C}_{5,1}$	15.42





Mwy, frame #492

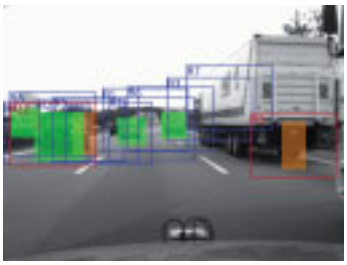


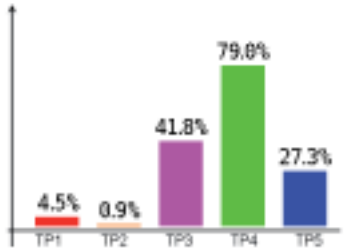
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.277
$\mathcal{R}_2$	3.717
$\mathcal{R}_3$	2.932
$\mathcal{R}_4$	3.538
$\mathcal{R}_5$	3.212
$\mathcal{R}_6$	3.556
$\mathcal{R}_7$	3.255
$\mathcal{R}_8$	3.655
$\mathcal{R}_9$	3.224
$\mathcal{R}_{10}$	3.710

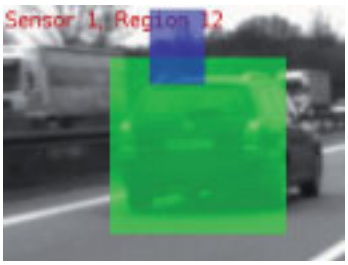
Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_2$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_2$	$\mathcal{C}_{5,1}$	15.42





Mwy, frame #650

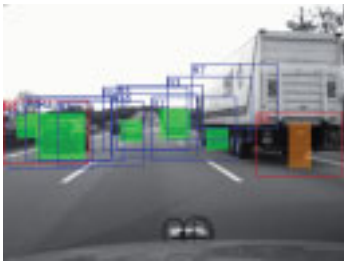


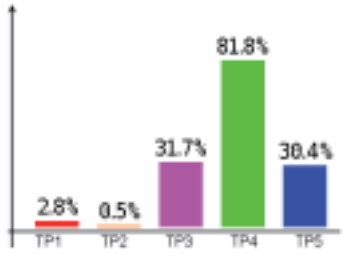
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.018
$\mathcal{R}_2$	3.328
$\mathcal{R}_3$	3.321
$\mathcal{R}_4$	3.094
$\mathcal{R}_5$	3.303
$\mathcal{R}_6$	2.953
$\mathcal{R}_8$	3.076
$\mathcal{R}_9$	3.539
$\mathcal{R}_{10}$	3.382
$\mathcal{R}_{11}$	2.992
$\mathcal{R}_{12}$	3.483

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_{12}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_9$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_{12}$	$\mathcal{C}_{5,1}$	15.42





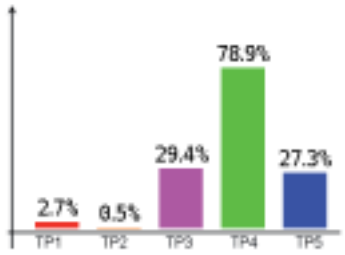
Mwy, frame #651

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.208
$\mathcal{R}_2$	3.167
$\mathcal{R}_3$	3.007
$\mathcal{R}_4$	3.441
$\mathcal{R}_5$	3.304
$\mathcal{R}_6$	3.157
$\mathcal{R}_8$	3.305
$\mathcal{R}_9$	3.494
$\mathcal{R}_{10}$	3.324
$\mathcal{R}_{11}$	3.400
$\mathcal{R}_{12}$	3.394

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_9$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_4$	$\mathcal{C}_{5,1}$	15.42



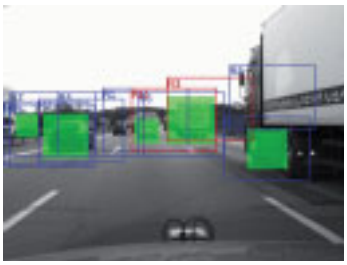
Mwy, frame #652

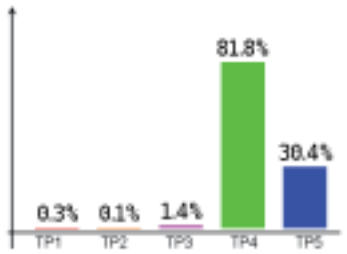
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	2.982
$\mathcal{R}_2$	3.364
$\mathcal{R}_3$	2.933
$\mathcal{R}_4$	3.571
$\mathcal{R}_5$	3.061
$\mathcal{R}_6$	3.977
$\mathcal{R}_8$	3.470
$\mathcal{R}_9$	3.495
$\mathcal{R}_{10}$	3.133
$\mathcal{R}_{11}$	3.994
$\mathcal{R}_{12}$	3.038

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_{11}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_6$	$\mathcal{C}_{1,1}$	16.36
3	$\mathcal{S}_2$	$\mathcal{R}_{11}$	$\mathcal{C}_{1,1}$	16.36





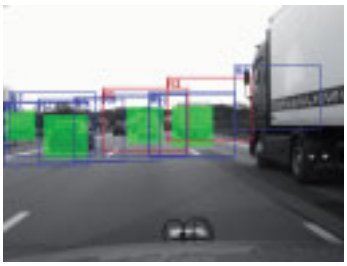
Mwy, frame #670

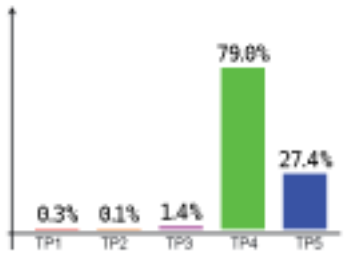
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.574
$\mathcal{R}_2$	3.426
$\mathcal{R}_3$	3.793
$\mathcal{R}_4$	3.596
$\mathcal{R}_5$	3.465
$\mathcal{R}_6$	3.483
$\mathcal{R}_7$	3.566
$\mathcal{R}_8$	3.293
$\mathcal{R}_9$	3.479
$\mathcal{R}_{10}$	3.026
$\mathcal{R}_{11}$	3.670

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_{11}$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_{11}$	$\mathcal{C}_{5,1}$	15.42





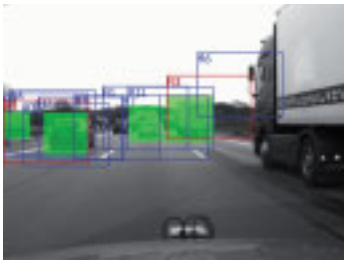
Mwy, frame #671

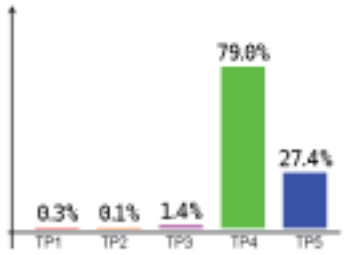
Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.691
$\mathcal{R}_2$	3.383
$\mathcal{R}_3$	3.832
$\mathcal{R}_4$	3.558
$\mathcal{R}_5$	3.321
$\mathcal{R}_6$	3.638
$\mathcal{R}_7$	3.481
$\mathcal{R}_8$	3.381
$\mathcal{R}_9$	3.653
$\mathcal{R}_{10}$	3.569
$\mathcal{R}_{11}$	3.649

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_2$	$\mathcal{R}_1$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_1$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_2$	$\mathcal{R}_1$	$\mathcal{C}_{5,1}$	15.42





Mwy, frame #672

Utility of candidate regions

$\mathcal{R}_n$	$\mathcal{U}^u(\mathcal{R}_n)$
$\mathcal{R}_1$	3.535
$\mathcal{R}_2$	3.456
$\mathcal{R}_3$	3.668
$\mathcal{R}_4$	3.548
$\mathcal{R}_5$	3.424
$\mathcal{R}_6$	3.286
$\mathcal{R}_7$	3.498
$\mathcal{R}_8$	3.515
$\mathcal{R}_9$	3.493
$\mathcal{R}_{10}$	3.345
$\mathcal{R}_{11}$	3.475

Classifier queue

$\mathcal{P}$	$\mathcal{S}$	$\mathcal{R}$	$\mathcal{C}$	$t_{\mathcal{C}}$ [ms]
1	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{4,1}$	22.87
2	$\mathcal{S}_2$	$\mathcal{R}_3$	$\mathcal{C}_{4,1}$	22.87
3	$\mathcal{S}_1$	$\mathcal{R}_4$	$\mathcal{C}_{5,1}$	15.42

Bibliography

[1] Eurostat. *Yearbook 2008*, chapter Transport. European Union, 2008.

[2] European Commision. Consultation paper on a 3rd road safety action plan 2002–2010. Technical report, European Commision, May 2001.

[3] Swedish Ministry of Transport and Communications. En route to a society with safe road traffic (ds 1997:13). Technical report, Swedish Ministry of Transport and Communications, 1997.

[4] P. Van Vliet and G. Schermers. Sustainable safety: A new approach for road safety in the netherlands. Technical report, Ministry of Transport, Public Works and Water Management; Traffic Research Centre, 2000.

[5] Marc Green, Merrill J. Allen, Bernard S. Abrams, and Leslie Weintraub. *Forensic Vision with Application to Highway Safety.* Lawyers & Judges Publishing, 3rd edition, 2008.

[6] Charles Anderson, David Van Essen, and Bruno Olshausen. *Neurobiology of Attention*, chapter Directed Visual Attention and the Dynamic Control of Information Flow, pages 11–17. Elsevier, 2005.

[7] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods, and applications. *Knowledge Engineering Review*, 11(2), 1996.

[8] Tom R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[9] Ling Liu and M. Tamer Özsu, editors. *Encyclopedia of Database Systems*, chapter Ontology. Springer, 2008.

[10] Arbeitsgruppe Umweltökonomische Gesamtrechnungen der Länder. Energieverbrauch und Treibhausgasemissionen (Energy consumption and greenhouse gas emission). Technical report, Statistische Ämter der Länder, 2007.

[11] Andreas Hermann, Stephan Matzka, and Joerg Desel. Using a proactive sensor-system in the distributed environment model. In *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium*, pages 703–708, 2008.

[12] Stephan Matzka, Yvan R. Petillot, and Andrew M. Wallace. Efficient resource allocation using a multiobjective utility optimisation method. In *ECCV Workshop on Multi-Camera and Multi-modal Sensor Fusion*, 2008.

[13] Stephan Matzka, Yvan R. Petillot, and Andrew M. Wallace. Fast motion estimation on range image sequences acquired with a 3-d camera. In *Proceedings of the British Machine Vision Conference*, volume II, pages 750–759. BMVA Press, 2007.

[14] Stephan Matzka, Yvan R. Petillot, and Andrew M. Wallace. Determining efficient scan-patterns for 3-d object recognition. In *Proceedings of the 3rd International Symposium on Visual Computing*, Lecture Notes in Computer Science 4842, pages 559–570. Springer, 2007.

[15] Stephan Matzka and Richard Altendorfer. A comparison of track-to-track fusion algorithms for automotive sensor fusion. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 189–194, Seoul, Korea, August 2008.

[16] Stephan Matzka and Richard Altendorfer. *Multisensor Fusion and Integration for Intelligent Systems*, chapter A Comparison of Track-to-Track Fusion Algorithms for Automotive Sensor Fusion, pages 69–82. Springer, 2009.

[17] James. L. Crowley. Active computer vision. Technical report, Grenoble INP, 1995.

[18] Dana H. Ballard and Christopher M. Brown. *Computer Vision*. Prentice Hall, 1992.

[19] David A. Forsyth and Jean Ponce. *Computer Vision: A modern approach*. Prentice Hall, 2002.

[20] Mosby. *Mosby's Medical Dictionary*. Elsevier, 2008.

[21] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, Jun 2001.

[22] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and BenWegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the 18th National Conference on Artificial Intelligence*, 2002.

[23] H. R. Everett. *Sensors for Mobile Robots: Theory and Application*. Peters, 1995.

[24] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, 14(4):231–249, 1997.

[25] Simone Frintrop, Erich Rome, Andreas Nüchter, and Hartmut Surmann. A bimodal laser-based attention system. *Computer Vision and Image Understanding*, 100(1-2): 124–151, 2005.

[26] P. Kohlhepp, P. Pozzo, M. Walther, and R. Dillmann. Sequential 3d-slam for mobile action planning. volume 1, pages 722–729, Sept.-2 Oct. 2004.

[27] Christian Brenneke, Oliver Wulf, and Bernardo Wagner. Autonome Navigation mobiler Systeme in natürlichen Umgebungen durch die Integration von 3D-Laserdaten - Autonomous navigation of mobile systems in natural environments by integration of 3d laser range data. *at - Automatisierungstechnik*, 53(2):59–69, 2005.

[28] E.D. Dickmanns, R. Behringer, C. Brudigam, D. Dickmanns, F. Thomanek, and V. van Holt. An all-transputer visual autobahn-autopilot/copilot. pages 608–615, May 1993.

[29] E.D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrandt, M. Maurer, F. Thomanek, and J. Schiehlen. The seeing passenger car 'vamors-p'. pages 68–73, Oct. 1994.

[30] R. Behringer and N. Muller. Autonomous road vehicle guidance from autobahnen to narrow curves. *Robotics and Automation, IEEE Transactions on*, 14(5):810–815, Oct 1998.

[31] DARPA. *Urban Challenge Rules*, October 2007.

[32] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, John Dolan, Dave Duggins, Dave Ferguson, Tugrul Galatali, Chris Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Tom Howard, Alonzo Kelly, David Kohanbash, Maxim Likhachev, Nick Miller, Kevin Peterson, Raj Rajkumar, Paul Rybski, Bryan Salesky, Sebastian Scherer, Young Woo-Seo, Reid Simmons, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William Whittaker, and Jason Ziglar. Tartan racing: A multi-modal approach to the DARPA urban challenge. Technical report, DARPA Urban Challenge, 2007.

[33] Sebastian Thrun, Burkhard Huhnke, Ganymed Stanek, Suhrid Bhat, Mike Montemerlo, Jesse Levinson, Anya Petrovskaya, Gabe Hoffmann, Doug Johnston, Dirk Hähnel, Dmitri Dolgov, Pamela Mahoney, David Orenstein, and Steve Keyes. Stanford's robotic vehicle Junior. Technical report, DARPA Urban Challenge, 2007.

[34] Charles Reinholtz, Thomas Alberi, David Anderson, Andrew Bacha, Cheryl Bauman, Stephen Cacciola, Patrick Currier, Aaron Dalton, Jesse Farmer, Ruel Faruque, Michael Fleming, Scott Frash, Grant Gothing, Jesse Hurdus, Shawn Kimmel, Chris Sharkey, Andrew Taylor, Chris Terwelp, David Van Covern, Mike Webster, and Al Wicks. Team Victor Tango: DARPA urban challenge technical paper. Technical report, DARPA Urban Challenge, 2007.

[35] Fred W. Rauskolb, Kai Berger, Christian Lipski, Marcus Magnor, Karsten Cornelsen, Jan Effertz, Thomas Form, Fabian Graefe, Sebastian Ohl, Walter Schumacher, Jörn-Marten Wille, Peter Hecker, Tobias Nothdurft, Michael Doering, Kai Homeier, Johannes Morgenroth, Lars Wolf, Christian Basarke, Christian Berger, Tim Gülke, Felix Klose, and Bernhard Rumpe. Caroline: An autonomously driving vehicle for urban environments. *Journal of Field Robotics*, 25(9):674–724, 2008.

[36] Jan Effertz. Sensor architecture and data fusion for robotic perception in urban environments at the 2007 darpa urban challenge. In *Robot Vision*, pages 275–290, 2008.

[37] Li Li and Fei-Yue Wang. *Advanced Motion Control and Sensing for Intelligent Vehicles*, chapter Intelligent Vehicle Vision Systems, pages 323–399. Springer, 2007.

[38] B. Elias and P. Mahonen. Pedestrian recognition based on 3d image data. In *Proceedings of the IEEE Industrial Symposium on Industrial Electronics*, pages 1406–1411, 2007.

[39] U. Scheunert, B. Fardi, N. Mattern, G. Wanielik, and N. Keppeler. Free space determination for parking slots using a 3d pmd sensor. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 154–159, 2007.

[40] P. Bergmiller, M. Botsch, J. Speth, and U. Hofmann. Vehicle rear detection in images with generalized radial-basis-function classifiers. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 226–233, 2008.

[41] Hartmut Surmann, Kai Lingemann, Andreas Nüchter, and Joachim Hertzberg. A 3d laser range finder for autonomous mobile robots. In *Proceedings of the 32nd ISR(International Symposium on Robotics)*, pages 153 – 158, 2001.

[42] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[43] Erik B. Sudderth, Antonio B. Torralba, William T. Freeman, and Alan S. Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77(1–3):291–330, 2008.

[44] Erik B. Sudderth, Antonio B. Torralba, William T. Freeman, and Alan S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *Proceedings of the ICCV*, pages 1331–1338, 2005.

[45] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, c-22(1):67–92, 1973.

[46] Rob Fergus, Pietro Perona, and Andrew Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3):273–303, 2007.

[47] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[48] Li Fei-Fei, Rob Fergus, and Antonio Torralba. Short course: Recognizing and learning object categories. In *Proceedings of the International Conference on Computer Vision*, 2005.

[49] Li Fei-Fei, Rob Fergus, and Antonio Torralba. Short course: Recognizing and learning object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[50] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, 1998.

[51] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[52] Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[53] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, 1st edition, 2008.

[54] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:148–156, 1997.

[55] Takeshi Mita, Toshimitsu Kaneko, Björn Stenger, and Osamu Hori. Discriminative feature co-orrucence selection for object detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(7):1257–1269, 2008.

[56] Franklin Crow. Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, number 207–212, 1984.

[57] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[58] B. Fardi, J. Dousa, G. Wanielik, B. Elias, and A. Barke. Obstacle detection and pedestrian recognition using a 3d PMD camera. In *IEEE Intelligent Vehicles Symposium*, 2006.

[59] S.D. Cochran and Gérard Medioni. 3-d surface description from binocular stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):981–994, 1992.

[60] Samuel Gidel, Christophe Blanc, Thierry Chateau, Paul Checchin, and Laurent Trassoudaine. A method based on multilayer laserscanner to detect and track pedestrians in urban environment. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 157–162, June 2009.

[61] Allesandro Chiuso, Paolo Favaro, Hailin Jin, and Stefano Soatto. Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535, 2002.

[62] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, volume 1, pages 44–57, 2008.

[63] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):406–417, 2005.

[64] Julian Ryde and Huosheng Hu. 3d laser range scanner with hemispherical field of view for robot navigation. In *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics*, pages 891–896, 2008.

[65] Paul J. Besl and Ramesh C. Jain. Three-dimensional object recognition. *ACM Comput. Surv.*, 17(1):75–145, 1985.

[66] Jan J. Koenderink and Andrea J. van Doorn. Surface shape and curvature scales. *Image Vision Comput.*, 10(8):557–565, 1992.

[67] Péter Csákány and Andrew M. Wallace. Representation and classification of 3-d objects. *IEEE Transactions on Sytems, Man, and Cybernetics Part B: Cybernetics*, 33(4):638 – 647, 2003.

[68] Fridtjof Stein and Gérard Medioni. Structural indexing: Efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2): 125–145, 1992.

[69] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[70] Richard J. Campbell and Patrick J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81 (2):166–210, 2001.

[71] Chin Seng Chua and Ray Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1997.

[72] James T. Reason. *Human error*. Cambridge University Press, 17. edition, 2006.

[73] Edward N. Zalta, editor. *The Stanford Encyclopedia of Philosophy*, chapter Risk, page http://stanford.library.usyd.edu.au/archives/fall2008/entries/risk/. Metaphysics Research Lab, Stanford University, Fall 2008 edition, 2008.

[74] John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, Princeton, 2nd ed. edition, 1947.

[75] Robert Nozick. *Anarchy, State, and Utopia*. Basic Books, 1974.

[76] T.M. Scanlon. *Utilitarianism and Beyond*, chapter Contractualism and Utilitarianism. Cambridge University Press, 1982.

[77] Diplomatic Conference for the Establishment of International Conventions for the Protection of Victims of War, editor. *Geneva Convention relative to the Protection of Civilian Persons in Time of War*. Office of the High Commissioner of Human Rights, 1949.

[78] Michael Walzer. *Just and Unjust Wars*. Basic Books, third edition, 1977.

[79] Council of Europe. *European Convention on Human Rights (Article 2)*. Council of Europe, 1950.

[80] Edward N. Zalta, editor. *The Stanford Encyclopedia of Philosophy*, chapter Doctrine of Double Effect, pages http://plato.stanford.edu/archives/win2008/entries/double–effect/. Metaphysics Research Lab, Stanford University, Winter 2008 edition, 2008.

[81] St. Thomas Aquinas. *The Summa Theologica*. Benziger Bros., fathers of the english dominican province translation edition, 1974.

[82] Joseph Mangan. An historical analysis of the principle of double effect. *Theological Studies*, 10:41–61, 1949.

[83] Yann Chevaleyre, Paul E. Dunne, Ulle Endriss, Jérôme Lang, Michel Lemaître, Nicolas Maudet, Julian Padget, Steve Phelps, Juan A. Rodríguez-Aguilar, and Paulo Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[84] Evangelos Triantaphyllou. *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer, 2000.

[85] Matthias Ehrgott. *Multicriteria Optimization*. Springer, 2nd edition, 2005.

[86] Vincent T. Kindt and Jean-Charles Billaut. *Multicriteria Scheduling. Theory, Models and Algorithms*. Springer, 2nd edition, 2006.

[87] John Nash. Non-cooperative games. *The Annals of Mathematics, 2nd Ser.*, 54(2): 286–295, 1951.

[88] Javier F. Seara. *Intelligent gaze control for vision-guided humanoid walking.* PhD thesis, Technische Universität München, 2004.

[89] Javier F. Seara and Günther Schmidt. Intelligent gaze control for vision-guided humanoid walking: methodological aspects. *Robotics and Autonomous Systems*, 48 (4):231–248, 2004.

[90] Carlos A. Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer, 2007.

[91] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *ECAI '96: Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, London, UK, 1997. Springer.

[92] Michael Wooldridge. *An Introduction to Multiagent Systems.* Wiley, 2nd edition, 2002.

[93] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[94] Thomas Pitz. *Anwendung Genetischer Algorithmen auf Handlungsbäume in Multiagentensystemen zur Simulation sozialen Handelns.* PhD thesis, Universität München, 2000.

[95] Tuomas W. Sandholm. Distributed rational decision making. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. The MIT Press, 1999.

[96] Jean-Pierre Aubin. *Optima and equilibria. An introduction to nonlinear analysis.* Springer, 2nd edition, 1998.

[97] E. Ephrati and J.S. Rosenschein. Divide and conquer in multi-agent planning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 375–380, 1994.

[98] Anne Treisman. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31(2):156–177, 1985.

[99] Laurent Itti, Christof Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov 1998.

[100] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506, May 2000.

[101] Laurent Itti. *Models of Bottom-Up and Top-Down Visual Attention.* PhD thesis, California Institute of Technology, Computation and Neural Systems, 2000.

[102] Simone Frintrop, Andreas Nüchter, Hartmut Surmann, and Joachim Hertzberg. Saliency-based object recognition in 3d data. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[103] Simone Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search.* LNAI 3899. Springer, 2006.

[104] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *In Proceedings of the European Conference on Coputer Vision*, volume 1, pages 228–241, 2004.

[105] Noel Trujillo, Roland Chapuis, Frederic Chausse, and Michel Naranjo. Object recognition: A focused vision based approach. In *Proceedings of the 3rd International Symposium on Visual Computing*, Lecture Notes in Computer Science 4842, pages 631–642. Springer, 2007.

[106] Ansgar Koene, Jan Morén, Vlad Trifa, and Gordon Cheng. Gaze shift reflex in a humanoid active vision system. In *Proceedings of the International Cognitive Vision Symposium Workshop*, 2007.

[107] Robert F. Schmidt and Hans-Georg Schaible. *Neuro- und Sinnesphysiologie – Neurophysiology*. Springer, 5th edition, 2006.

[108] Robert F. Schmidt and Florian Lang. *Physiologie des Menschen – Human Physiology*. Springer, 30th edition, 2007.

[109] John M. Henderson. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences*, 7(11):498–504, 2003.

[110] Wolfgang Einhäuser, Wolfgang Kruse, Klaus-Peter Hoffmann, and Peter König. Differences of monkey and human overt attention under natural conditions. *Vision Research*, 46(8-9):1194–1209, 2006.

[111] Mark S. Nixon and Alberto S. Aguado. *Feature Extraction and Image Processing*. Newnes, Oxford, first edition, 2002.

[112] David C. Van Essen, Bruno Olshausen, Charles Anderson, and J.L. Gallant. Pattern recognition, attention, and information bottlenecks in the primate visual system. In *Proceedings, SPIE Conference on Visual Information Processing: From Neurons to Chips*, volume 1473, pages 17–28, 1991.

[113] David C. van Essen and Charles H. Anderson. *An Introduction to Neural and Electronic Networks*, chapter Information processing strategies and pathways in the primate visual system, pages 45–76. Academic Press, 2nd edition, 1995.

[114] C. Eliasmith and Charles Anderson. *Neural Engineering*. MIT Press, 2003.

[115] T.K. Landauer. How much do people remember? some estimates of the quantitiy of learned information in long-term memory. *Cognitive Science*, 10:477–493, 1986.

[116] R. H. Masland. The fundamental plan of the retina. *Nature Neuroscience*, 4(9): 877–886, 2001.

[117] David H. Hubel. *Eye, Brain, and Vision*. Scientific American, 1995.

[118] Kevin N. Walker, Timothy F. Cootes, and Christopher J. Taylor. Locating salient object features. In *British Machine Vision Conference (BMVC)*, pages 557–566. BMVA Press, 1998.

[119] J.P. Collomosse and P.M. Hall. Cubist style rendering from photographs. *IEEE Transactions on Visualisation and Computer Graphics*, 9(4):443–453, 2003.

[120] Pierre Baldi. *Information, Coding, and Mathematics*, chapter A computational theory of surprise, pages 1–25. Kluwer Academic, 2002.

[121] Pierre Baldi. *Neurobiology of Attention*, chapter Surprise: A shortcut for Attention?, pages 24–28. Elsevier, 2005.

[122] P.J. Burt and E.H. Adelson. The laplacian pyramid as a compart image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[123] Dennis Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers*, 93(3):429–459, 1946.

[124] A.G. Leventhal. *The Neural Basis of Visual Function (Vision and Visual Dysfunction Vol. 4)*. CRC Press, 1991.

[125] S. Engel, X. Zhang, and B. Wandell. Cplour tuning in human visual cortex measured with functional magnetic resonance imaging. *Nature*, 388(6637):68–71, 1997.

[126] R.L. DeValois, Albrecht D.G., and L.G. Thorell. Spatial-frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559, 1982.

[127] R. Tootell, S.L. Hamilton, M.S. Silverman, and E. Switkes. Functional anatomy of macaque striate cortex. i. ocular dominance, binocular interactions, and baseline conditions. *Journal of Neuroscience*, 8(5), 1988.

[128] K. Arai and E.L. Keller. A model of the saccade-generating system that accounts for trajectory variations produced by competing visual stimuli. *Biological Cybernetics*, 92:21–37, 2005.

[129] D.K. Lee, Laurent Itti, and J. Braun. Attention activates winner-take-all competition among visual filters. *Nature Neuroscience*, 2:375–381, 1999.

[130] Christof Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.

[131] Zhaoping Li. A saliency map in primary visual cortex. *TRENDS in Cognitive Sciences*, 6(1):9–16, 2002.

[132] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[133] Timor Kadir. *Scale, Saliency and Scene Description*. PhD thesis, University of Oxford, Department of Engineering Science, 2002.

[134] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10): 1615–1630, 2005.

[135] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.

[136] Vidhya Navalpakkam and Laurent Itti. Optimal cue selection strategy. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 987–994. MIT Press, Cambridge, MA, 2006.

[137] Vidhya Navalpakkam and Laurent Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 02, pages 2049–2056, 2006.

[138] Simone Frintrop, Gerriet Backer, and Erich Rome. Goal-directed search with a top-down modulated computational attention system. In *Proceedings of the Annual Meeting of the German Association for Pattern Recognition*, 2005.

[139] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.

[140] Noel Trujillo, Roland Chapuis, Frederic Chausse, and C. Blanc. On road simultaneous vehicle recognition and localization by model based focused vision. In *Proceedings of the IAPR Conference on Machine Vision Applications*, 2005.

[141] Stefan Treue and Julio C. Martinez-Trujillo. Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399:575–579, 1999.

[142] Julio C. Martinez-Trujillo and Stefan Treue. Feature-based attention increases the selectivity of population responses in primate visual cortex. *Current Biology*, 14: 744–751, 2004.

[143] Jeremy M. Wolfe. Guided search 4.0: A guided search model that does not require memory for rejected distractors. *Journal of Vision*, 1(3):349–349, 2001.

[144] Sara Mitri, Simone Frintrop, Kai Pervölz, Hartmut Surmann, and Andreas Nüchter. Robust object detection at regions of interest with an application in ball recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 125–130, 2005.

[145] J. Theeuwes. Top-down search strategies cannot override attentional capture. *Psychonomic Bulletin & Review*, 11:65–70, 2004.

[146] Antonio Torralba, Aude Oliva, Monica S. Castelhano, and John M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search. *Psychological Review*, 113(4):766–786, 2006.

[147] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[148] Aude Oliva and Antonio Torralba. Building the gist of a scene: the role of global image features in recognition. *Progress in Brain Research*, Special Issue on Visual Perception, 2006.

[149] Javier F. Seara, K. H. Strobl, and G. Schmidt. Information management for gaze control in vision guided biped walking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 31–36, 2002.

[150] C.E. Shannon. A mathematical theory of communication. Technical report, Bell System Technical Journal, 1948.

[151] Michael Argyle. *Bodily Communication*. Routledge, 2nd edition, 1988.

[152] Guochang Xu. *GPS: Theory, Algorithms and Applications*. Springer, 2007.

[153] *Technical specifications - AXIS 231D+/232D+ Network Dome Cameras*. Axis Communications, 2008.

[154] Nai-Xiang Lian, Lanlan Chang, Vitali Zagorodnov, and Yap-Peng Tan. Reversing demosaicking and compression in color filter array image processing: Performance analysis and modeling. *IEEE Transactions on Image Processing*, 15(11):3261–3278, 2006.

[155] Cang Ye and Johann Borenstein. Characterization of a 2-d laser scanner for mobile robot obstacle negotiation. In *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pages 2512–2518. IEEE, 2002.

[156] ISO/TC 22 Road vehicles. ISO 8855:1991–Road vehicles; vehicle dynamics and road-holding ability; vocabulary. ISO International Organization for Standardization, 1991.

[157] K. F. Riley, M. P. Hobson, and S. J. Bence. *Mathematical Methods for Physics and Engineering*. Cambridge University Press, 3rd edition, 2006.

[158] Ingrid Carlbom and Joseph Paciorek. Planar geometric projections and viewing transformations. *ACM Computing Surveys*, 10(4):465–502, 1978.

[159] Matt Pharr and Greg Humphreys. *Physically Based Rendering. From Theory to Implementation*, chapter 7: Sampling and Reconstruction, pages 279–367. Morgan Kaufmann, 2004.

[160] Alexis Michael Tourapis, Oscar C. Au, and Ming Lei Liou. Predictive motion vector field adaptive search technique (PMVFAST) - enhancing block based motion estimation. In *Proceedings of Visual Communications and Image Processing*, 2001.

[161] D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9: 272–290, 1997.

[162] Krishnendu Chaudhury, Rajiv Mehrotra, and Cid Srinivasan. Detecting 3-d motion field from range image sequences. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 29(2):308–314, 1999.

[163] H. Spies, B. Jähne, and J. L. Barron. Range flow estimation. *Computer Vision Image Understanding*, 85(3):209–231, 2002.

[164] Yonghuai Liu and Marcos A. Rodrigues. Correspondenceless motion estimation from range images. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, volume 1, pages 654–660. IEEE Computer Society, 1999.

[165] X. Jiang, S. Hofer, T. Stahs, I. Ahrns, and H. Bunke. Extraction and tracking of surfaces in range image sequences. In *Proceedings of the 2nd International Conference on 3-D Digital Imaging and Modeling*, pages 252–260, 1999.

[166] P.I. Hosur and K.K. Ma. Motion vector field adaptive fast motion estimation. In *Second International Conference on Information, Communications and Signal Processing (ICICS'99), Singapore*, 1999.

[167] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dec 2000.

[168] Mann El Badaoui El Najjar and Philippe Bonnifait. A road-matching method for precise vehicle localization using belief theory and kalman filtering. *Autonomous Robots*, 19(2):173–191, 2005.

[169] Mann El Badaoui El Najjar and Philippe Bonnifait. Road selection using multi-criteria fusion for the road-matching problem. *IEEE Transactions on Intelligent Transportation Systems*, 8:279–299, 2007.

[170] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11): 1863–1868, 2006.

[171] F. Blais, J.A. Beraldin, S.F. El-Hakim, and L. Cournoyer. Real-time geometrical tracking and pose estimation using laser triangulation and photogrammetry. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 205–210, 2001.

[172] F.R. Hampel. A general qualitative definition of robustness. *Annals of Mathematical Statistics*, 42:1887–1896, 1971.

[173] P.J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.

[174] P.J. Bickel. One-step Huber estimates in the linear model. *Journal of the American Statistical Association*, 70:428–434, 1975.

[175] P.J. Rousseeuw and K. Van Driessen. *Data Analysis: Scientific Modeling and Practical Application*, chapter An Algorithm for Positive-Breakdown Regression Based on Concentration Steps, pages 335–346. Springer, 2000.

[176] S. Verboven and M. (2005) Hubert. LIBRA: a MATLAB library for robust analysis. *Chemometrics and Intelligent Laboratory Systems*, 75:127–136, 2005.

[177] D. N. Joanes and C. A. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998.

[178] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[179] Samuel Blackman and Robert Popoli. *Design and analysis of modern tracking systems*. Artech House radar library. Artech House, 2nd edition, 1999.

[180] Richard van der Horst and Jeroen Hogema. Time-to-collision and collision avoidance systems. In *Proceedings of the 6th ICTCT-Workshop*, 1994.

[181] J.Ch. Hayward. TTSC 7115: Near miss determination through use of a scale of danger. Technical report, Pennsylvania State University, 1972.

[182] John M. Galbraith, Garrett T. Kenyon, and Richard W. Ziolkowski. Time-to-collision estimation from motion based on primate visual processing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1279–1291, 2005.

[183] M. Wegscheider and G. Prokop. Modellbasierte komfortbewertung von fahrerassistenzsystzemen. In *Proceedings of 7. Symposium Numerische Simulation in der Fahrzeugtechnik*, volume 1900 of *VDI Berichte*, pages 17–36, 2005.

[184] H. Mori and N.M. Charkari. Shadow and rhythm as sign patterns of obstacle detection. pages 271–277, 1993.

[185] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, May 2006.

[186] Statistisches Bundesamt. *Verkehrsunfälle 2006 (Survey of road accidents in 2006)*. Fachserie 8, Reihe 7. Statistisches Bundesamt, Wiesbaden, revised edition, Nov. 2007.

[187] Jeffrey K. Uhlmann. Covariance consistency methods for fault-tolerant distributed data fusion. *Information Fusion*, 4(3):201–215, 2003.

[188] Simon J. Julier and Jeffrey K. Uhlmann. *Handbook of Data Fusion*, chapter 12: General decentralized data fusion with covariance intersection (CI), pages 1–25. CRC Press, 2001.

[189] S.P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[190] H. Steinhaus. Sur la division des corp materiels en parties. *Bulletin L'Acadmie Polonaise des Science, C1. III*, IV:245–269, 1956.

[191] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:881–892, 2002.

[192] Thomas H. Cormen, Charles E. Leiserson, , and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 2000.

[193] D.H. Stamatis. *Failure mode and effect analysis*. Quality Press, 2nd edition, 2003.

[194] Dyadem Press, editor. *Guidelines for Failure Mode and Effects Analysis for Automotive, Aerospace, and General Manufacturing Industries*. CRC Press, 2003.

[195] Dyadem Press, editor. *Guidelines for Failure Modes and Effects Analysis for Medical Devices*. CRC Press, 2003.

[196] P.L. Goddard. Software FMEA techniques. In *Proceedings of the Annual Reliability and Maintainability Symposium*, pages 118–123, 2000.

[197] Karl-Josef Höhnscheid and Martina Straube. *Socio-economic costs due to road traffic accidents in Germany 2004*. Bundesanstalt für Straßenwesen, Bergisch Gladbach, 2006.

[198] Matthias Kühn, Robert Fröming, and Volker Schindler. *Fußgängerschutz - Pedestrian Safety*. Springer, 2007.

[199] Stephan Kopischke, Katharina Seifert, and Maria Hoppe. Möglichkeiten und grenzen von kollisionswarnsystemen. In *Proceedings of 25. Tagung Integrierte Sicherheit und Fahrerassistenzsysteme*, 2008.

[200] Florian Mühlfeld, Johannes Happe, and Thomas Brandmeier. Situation analysis for pre-crash-safety-systems - model for the evalution of the traffic criticality level. In *Proceedings of 14. Kongress Elektronik im Kraftfahrzeug*, 2009.

[201] Nils Krahnstoever, Ting Yu, Ser-Nam Lim, Kedar Patwardhan, and Peter Tu. Collaborative real-time control of active cameras in large scale surveillance systems. In *ECCV Workshop on Multi-Camera and Multi-modal Sensor Fusion*, 2008.

[202] Leonard Eugene Dickson. *History of the Theory of Numbers*, volume II: Diophantine Analysis. Dover Publications, 2005.

[203] Robert R. Fenichel. Algorithms: Algorithm 329: Distribution of indistinguishable objects into distinguishable slots. *Commununications of the ACM*, 11(6):430, 1968.

[204] M. Gray. Remark on algorithm 329 [g6]: distribution of indistinguishable objects into distinguishable slots. *Commununications of the ACM*, 12(3):187, 1969.

[205] Paul Viola and William M. Wells. Alignment by maximization of mutual information. In *Proceedings of IEEE International Conference on Computer Vision*, pages 16–23, 1995.

[206] Paul Viola and William M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[207] A. Collignon, F. Maes, D. Delaere, Vandermeulen D., P. Suetens, and G. Marchal. *Information Processing in Medical Imaging*, chapter Automated multimodality image registration based on information theory, pages 263–274. Kluwer, 1995.

[208] Stephen Gould, Paul Baumstarck, Morgan Quigley, Andrew Y. Ng, and Daphne Koller. Integrating visual and range data for robotic object detection. In *ECCV Workshop on Multi-Camera and Multi-modal Sensor Fusion*, 2008.

[209] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981.

[210] Ronald Graham, Donald Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 2nd edition, 1994.

[211] Joseph Y.-T. Leung, editor. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapmann & Hall/CRC, 2004.

[212] William Stallings. *Operating Systems: Internals and Design Principles*. Prentice Hall, 5th edition, 2005.

[213] Kirk Pruhs, Jirí Sgall, and Eric Torng. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 15: Online Scheduling, pages 15.1–15.43. Chapmann & Hall/CRC, 2004.

[214] John Rawls. *A Theory of Justice*. Harvard University Press, revised edition, 1999.

[215] Stephan Matzka, Yvan R. Petillot, Andrew M. Wallace, and Paul Sprickmann Kerkerinck. Increasing sensor-resource efficiency using a proactive sensor system. In *3rd Conference "Active Safety through Driver Assistance"*, pages 26.1–26.4, 2008.

[216] Gunter Magin, Achim Ruß, Darius Burschka, and Georg Faerber. A dynamic 3d environmental model with real-time access functions for use in autonomous mobile robots. *Robotics and Autonomous Systems*, 14:119–131, 1995.