Ensuring the Take-Over Readiness of the Driver Based on the Gaze Behavior in Conditionally Automated Driving Scenarios

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Christian Braunagel

aus Ravensburg

Tübingen
2017

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	09.03.2018
Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Wolfgang Rosenstiel
2. Berichterstatter:	Jun.-Prof. Dr. Enkelejda Kasneci

" All men dream, but not equally.

Those who dream by night in the dusty recesses of their minds,

wake up in the day to find it was vanity:

but the dreamers of the day are dangerous men,

for they may act their dreams with open eyes, to make it possible.

T. E. Lawrence

Acknowledgments

Diese Arbeit entstand während meiner fast dreieinhalb-jährigen Doktorandenzeit bei der Daimler AG in Sindelfingen. Diese spannende und vor allem anspruchsvolle Zeit hat mich sowohl beruflich als auch persönlich geprägt wie keine andere Zeit in meinem Leben und wird mir deshalb stets in Erinnerung bleiben. Dasselbe gilt aber auch für die Menschen, die ich in dieser Zeit kennenlernen durfte und die mich über die gesamte Dauer meiner Arbeit unentwegt unterstützt haben.

Deshalb möchte ich mich zunächst bei meinem Doktorvater und ersten Berichterstatter Prof. Dr. Wolfgang Rosenstiel von der Universität Tübingen bedanken. Trotz seiner zeitintensiven Aufgaben als Dekan der mathematisch-naturwissenschaftlichen Fakultät hatte er stets ein offenes Ohr für fachliche Diskussionen, in denen er mit konstruktiven Argumenten die Arbeit immer weiter vorangetrieben hat. Ein ganz besonderer Dank gilt meiner Zweitgutachterin und Betreuerin Prof. Dr. Enkelejda Kasneci, die mir von Beginn an und zu wirklich jeder Zeit mit Rat und Tat zur Seite stand. Ohne Ihre Ratschläge und fachliche Expertise sowie Ihre hilfreichen Reviews wäre diese Arbeit so nicht realisierbar gewesen.

Natürlich gilt auch meinen Kollegen hier bei der Daimler AG mein Dank. An erster Stelle geht dieser Dank an meinen Betreuer Dr. Wolfgang Stolzmann, den ich schon in meiner Zeit als Bachelorstudent kennenlernen durfte. Er war es, der sich für diese Stelle eingesetzt und diese Promotion überhaupt erst ermöglicht hat. Zudem war seine langjährige Erfahrung im Bereich Fahrerbeobachtung sowie bei der Betreuung von Versuchsreihen mir stets eine willkommene Unterstützung. Ein herzliches Dankeschön auch an alle meine Kollegen, mit denen ich all die Jahre hinweg beruflich und privat so viele interessante und lustige Momente erleben durfte. Ich finde man kann sich glücklich schätzen, wenn man jeden Tag nicht nur mit Kollegen, sondern mit Freunden zusammenarbeiten darf. Ein besonderer Dank geht dabei noch an meine ehemaligen Studenten David Geisler und Philipp Winter, für die großartige Zusammenarbeit und ihre erfolgreichen Abschlussarbeiten, die verschiedene Aspekte dieser Arbeit weiter vorangetrieben haben.

Eine wissenschaftliche Promotion erfordert aber nicht nur Unterstützung von fachlicher Seite, sondern auch aus dem privaten Umfeld. Gerade in zehrenden Zeiten, in denen man sich voll und ganz der wissenschaftlichen Arbeit hingeben muss, benötigt man das Verständnis und den Rückhalt seiner Familie und seines Freundeskreises. Darum möchte ich an dieser Stelle all meinen Freunden danken, die mich in dieser Zeit angespornt und motiviert haben, mir aber auch die Notwendigkeit des ein oder anderen Gin Tonics verdeutlicht haben. Ich möchte meinen Eltern, Emma und Viktor, danken, die mein Studium überhaupt erst ermöglicht haben, die immer für mich da waren, wenn ich sie gebraucht habe und vor allem für ihr entgegengebrachtes Vertrauen, diesen anspruchsvollen Weg erfolgreich zu beschreiten. Ein weiteres Dankeschön geht an meine Großeltern, meine Schwester und Ihre Familie für ihre Nachsicht und Geduld gerade in den letzten Jahren in denen ich zu wenig Zeit mit Ihnen verbracht habe. Zuletzt möchte ich dir, liebe Manuela, danken: dafür, dass du mir den Rücken gestärkt und freigehalten hast, Verständnis aufgebracht hast für die leider viel zu seltenen und zu kurzen gemeinsamen Urlaube und Aktivitäten und für deine ganze Liebe. Nun ist endlich wieder mehr Zeit dafür, gemeinsam diese spannende Welt zu erkunden.

Stuttgart, den 01.März 2017 *Christian Braunagel*

Abstract

Conditional automation is the next step towards the fully automated vehicle. Under prespecified conditions an automated driving function can take-over the driving task and the responsibility for the vehicle, thus enabling the driver to perform secondary tasks. However, performing secondary tasks and the resulting reduced attention towards the road may lead to critical situations in take-over situations. In such situations, the automated driving function reaches its limits, forcing the driver to take-over responsibility and the control of the vehicle again. Thus, the driver represents the fallback level for the conditionally automated system. At this point the question arises as to how it can be ensured that the driver can take-over adequately and timely without restricting the automated driving system or the new freedom of the driver.

To answer this question, this work proposes a novel prototype for an advanced driver assistance system which is able to automatically classify the driver's take-over readiness for keeping the driver ”in-the-loop”. The results show the feasibility of such a classification of the take-over readiness even in the highly dynamic vehicle environment using a machine learning approach. It was verified that far more than half of the drivers performing a low-quality take-over would have been warned shortly before the actual take-over, whereas nearly 90% of the drivers performing a high-quality take-over would not have been interrupted by the driver assistance system during a driving simulator study.

The classification of the take-over readiness of the driver is performed by means of machine learning algorithms. The underlying features for this classification are mainly based on the head and eye movement behavior of the driver. It is shown how the secondary tasks currently being performed as well as the glances on the road can be derived from these measured signals. Therefore, novel, online-capable approaches for driver-activity recognition and Eyes-on-Road detection are introduced, evaluated, and compared to each other based on both data of a simulator and real-driving study. These novel approaches are able to deal with multiple challenges of current state-of-the-art methods such as: i) only a coarse separation of driver activities possible, ii) necessity for costly and time-consuming calibrations, and iii) no adaption to conditionally automated driving scenarios.

Zusammenfassung

Das hochautomatisierte Fahren bildet den nächsten Schritt in der Evolution der Fahrerassistenzsysteme hin zu vollautomatisierten Fahrzeugen. Unter definierten Bedingungen kann dabei der Fahrer die Fahraufgabe inklusive der Verantwortung über das Fahrzeug einer automatisierten Fahrfunktion übergeben und erhält die Möglichkeit sich anderen Tätigkeiten zu widmen. Um dennoch sicherzustellen, dass der Fahrer bei Bedarf schnellstmöglich die Kontrolle über das Fahrzeug wieder übernehmen kann, stellt sich die Frage, wie die fehlende Aufmerksamkeit gegenüber dem Straßenverkehr kompensiert werden kann ohne dabei die hochautomatisierte Fahrfunktion oder die neu gewonnenen Freiheiten des Fahrers zu beschränken.

Um diese Frage zu beantworten wird in der vorliegenden Arbeit ein erstes prototypisches Fahrerassistenzsystem vorgestellt, welches es ermöglicht, die Übernahmebereitschaft des Fahrers automatisiert zu klassifizieren und abhängig davon den Fahrer "in-the-loop" zu halten. Die Ergebnisse zeigen, dass eine automatisierte Klassifikation über maschinelle Lernverfahren selbst in der hochdynamischen Fahrzeugumgebung hervorragende Erkennungsraten ermöglicht. In einer der durchgeführten Fahrsimulatorstudien konnte nachgewiesen werden, dass weit mehr als die Hälfte der Probanden mit einer geringen Übernahmequalität kurz vor der eigentlichen Übernahmesituation gewarnt und nahezu 90% der Probanden mit einer hohen Übernahmequalität in ihrer Nebentätigkeit nicht gestört worden wären.

Diese automatisierte Klassifizierung beruht auf Merkmalen, die über Fahrerbeobachtung mittels Innenraumkamera gewonnen werden. Für die Extraktion dieser Merkmale werden Verfahren zur Fahreraktivitätserkennung und zur Detektion von Blicken auf die Straße benötigt, welche aktuell noch mit gewissen Schwachstellen zu kämpfen haben wie:
i) Nur eine grobe Unterscheidung von Tätigkeiten möglich, ii) Notwendigkeit von kosten- und zeitintensiven Kalibrationsschritten, iii) fehlende Anpassung an hochautomatisierte Fahrszenarien. Aus diesen Gründen wurden neue Verfahren zur Fahreraktivitätserkennung und zur Detektion von Blicken auf die Straße in dieser Arbeit entwickelt, implementiert und evaluiert. Dabei bildet die Anwendbarkeit der Verfahren unter realistischen Bedingungen im Fahrzeug einen zentralen Aspekt. Zur Evaluation der einzelnen Teilsysteme und des übergeordneten Fahrerassistenzsystems wurden umfangreiche Versuche in einem Fahrsimulator sowie in realen Messfahrzeugen mit Referenz- sowie seriennaher Messtechnik durchgeführt.

Contents

Notation and Abbreviations

Notations

x	scalar value
$\boldsymbol{x}$	column vector
$\boldsymbol{X}$	Matrix
$[x]$	Interval variable

Mathematical Functions

$\lvert ... \rvert$	Cardinality in case of a set or the absolute value in case of a number
$\lvert\lvert ... \rvert\rvert$	Norm representing the Euclidean distance
$arccos$	Inverse function of the cosine
$arcsin$	Inverse function of the sine
$arctan$	Inverse function of the tangent
$atan2$	Inverse function of the tangent with two input parameters
cos	Function of the cosine
sin	Function of the sine
$\log_2$	Binary logarithm
max	Maximum operator
$E[...]$	Expectation value
$IG(...)$	Function of the information gain
$SU(...)$	Function of the symmetrical uncertainty correlation
$\sum$	Sum operator
$\mathcal{N}(...)$	Gaussian distribution
$O(...)$	Big O notation

Symbols

$\mathbf{lb}_{init}$	Origin of the *laserBird* coordinate system with regard to the vehicle coordinate system, $\mathbb{R}^{3\times1}$
$\mathbf{lb}_{pos}$	Head position vector in millimeter, $\mathbb{R}^{3\times1}$
lb_{pos_x}	Head position in x direction given in millimeter
lb_{pos_y}	Head position in y direction given in millimeter
lb_{pos_z}	Head position in z direction given in millimeter

$\mathbf{lb}_{pos,in}$	Head position vector in inches, $\mathbb{R}^{3\times 1}$
$lb_{pos_x,in}$	Head position in x direction given in inches
$lb_{pos_y,in}$	Head position in y direction given in inches
$lb_{pos_z,in}$	Head position in z direction given in inches
$\mathbf{lb}_{rot}$	Head rotation vector in radian, $\mathbb{R}^{3\times 1}$
ϕ_{lb}	Head rotation about the x axis of the *laserBird* given in radian
θ_{lb}	Head rotation about the y axis of the *laserBird* given in radian
ψ_{lb}	Head rotation about the z axis of the *laserBird* given in radian
$\mathbf{lb}_{rot,deg}$	Head rotation vector in degree, $\mathbb{R}^{3\times 1}$
$\phi_{lb,deg}$	Head rotation about the x axis given in degree
$\theta_{lb,deg}$	Head rotation about the x axis given in degree
$\psi_{lb,deg}$	Head rotation about the x axis given in degree
$[pmb_x]$	Interval of the x value of the Performance Motion Box
$[pmb_y]$	Interval of the y value of the Performance Motion Box
$[pmb_z]$	Interval of the z value of the Performance Motion Box
x_{gaze}	X value of a gaze direction in a cartesian coordinate system
y_{gaze}	Y value of a gaze direction in a cartesian coordinate system
z_{gaze}	Z value of a gaze direction in a cartesian coordinate system
$\mathbf{\Phi}_{gaze,cart}$	Gaze vector in a cartesian coordinate system, $\mathbb{R}^{3\times 1}$
θ	Pitch angle of a gaze vector in a spherical coordinate system
ψ	Yaw angle of a gaze direction in a spherical coordinate system
$\mathbf{\Phi}$	Gaze vector in a spherical coordinate system, $\mathbb{R}^{2\times 1}$
eye_x	Horizontal coordinate of the Dikablis eye camera in pixel
eye_y	Vertical coordinate of the Dikablis eye camera in pixel
$\mathbf{\Phi}_{DK}$	2-dimensional pixel coordinate of the Dikablis eye camera, $\mathbb{R}^{2\times 1}$
f	Function to transform pixel coordinates in spherical angles
$\mathbf{\Phi}_{eye}$	Vector describing the rotation of the eyes in a spherical coordinate system, $\mathbb{R}^{2\times 1}$
θ_{eye}	Pitch angle of the rotation of the eyes in a spherical coordinate system
ψ_{eye}	Yaw angle of the rotation of the eyes in a spherical coordinate system
$\mathbf{lb}_{nose}$	Distance between the *laserBird* head sensor and the subject's nosebridge, $\mathbb{R}^{3\times 1}$
$\mathcal{V}_{\text{ref}}$	Target used for the calibration step, $\mathbb{R}^{3\times 1}$
$q^2_{j,l}$	Squared loading of the j-th variable on the l-th factor
$\bar{q}^2_{j,l}$	Mean of the squared loadings
$\mathbf{lb}'_{pos}$	Head position vector in the vehicle coordinate system, $\mathbb{R}^{3\times 1}$
$\mathbf{R}_x$	Basic rotation matrix about the x axis, $\mathbb{R}^{3\times 3}$
$\mathbf{R}_y$	Basic rotation matrix about the y axis, $\mathbb{R}^{3\times 3}$
$\mathbf{R}_z$	Basic rotation matrix about the z axis, $\mathbb{R}^{3\times 3}$
$\mathbf{R}_{zyx}$	Combined rotation matrix with the order ZYX, $\mathbb{R}^{3\times 3}$
$\mathbf{lb}'_{rot}$	Head rotation vector of the *laserBird* with inverted axis and interchanged roll and pitch angle, $\mathbb{R}^{3\times 1}$
ϕ_{veh}	Head rotation about the x axis of the vehicle coordinate system

θ_{veh}	Head rotation about the y axis of the vehicle coordinate system
ψ_{veh}	Head rotation about the z axis of the vehicle coordinate system
$\mathbf{lb}_{origin}$	Origin of the head pose, $\mathbb{R}^{3\times 1}$
$\mathbf{lb}_{dir}$	Direction of the head pose, $\mathbb{R}^{3\times 1}$
$\mathbf{\Phi}_{lb}$	Head pose, $\mathbb{R}^{3\times 1}$
$\mathbf{\Phi}_{eye,cart}$	Vector describing the rotation of the eyes in a cartesian coordinate system, $\mathbb{R}^{2\times 1}$
$e\hat{y}e_x$	Horizontal coordinate of the Dikablis eye camera in pixel scaled to the interval $[-1,1]$
$e\hat{y}e_y$	Vertical coordinate of the Dikablis eye camera in pixel scaled to the interval $[-1,1]$
f^{-1}	Inverse function of f for transforming spherical angles to Dikablis pixel coordinates
$\mathbf{dk}_{dir}$	Direction of the eye gaze without the head pose, $\mathbb{R}^{3\times 1}$
p_f	Propability density function of the velocity profile of fixations
p_s	Propability density function of the velocity profile of saccades
p	Propability density function of the GMM for eye movement velocity profile
μ_f	Mean value of the distribution of the velocity profile of fixations
μ_s	Mean value of the distribution of the velocity profile of saccades
β_f	Variance of the distribution of the velocity profile of fixations
β_s	Variance of the distribution of the velocity profile of saccades
π_f	Mixture value of the distribution of the velocity profile of fixations
π_s	Mixture value of the distribution of the velocity profile of saccades
v_i	Eye movement velocity for the sample i
Θ_f	Parameter set of the distribution of the velocity profile of fixations
Θ_s	Parameter set of the distribution of the velocity profile of saccades
δ	Intersection point of the distributions of the velocity profile of fixations and saccades
ω	Weighting factor of the sample mean and sample variance
$\mathcal{Z}$	Random variable
V_{n+1}	Set of $n+1$ sequential eye velocities
$\tilde{\Theta}$	Estimated parameter set
l	Threshold for interchanging the actual with the estimated parameter
$\Theta_{f,init}$	Initial values of the parameter set for the distribution of the velocity profile of fixations
$\Theta_{s,init}$	Initial values of the parameter set for the distribution of the velocity profile of saccades
g	Hedges' g measure
p	Significance value
z	z Value
$p^{(w)}$	Probability density function of the cluster *windshield*
$p^{(hh)}$	Probability density function of the cluster *handheld*

$p^{(hf)}$	Probability density function of the cluster *hands-free*
$p^{(u)}$	Probability density function of the cluster *unknown*
χ	Set of learned clusters
φ_{eucl}	Euclidean distance of the pitch and yaw angle of the head rotation
g_i	Filtered output of the Savitzky-Golay filter
c_n	Coefficients for the first derivative of the Savitzky-Golay filter
n_L	Left half of the moving window of the Savitzky-Golay filter
n_R	Right half of the moving window of the Savitzky-Golay filter
h	Function of a first degree polynomial
$\varepsilon_{amp,init}$	Constant minimum angle threshold for EoR event detection
$\varepsilon_{vel,init}$	Constant minimum velocity threshold for EoR event detection
$\varepsilon_{amp,75}$	Learned 75%-quantile amplitude threshold for EoR plausibility check
$\varepsilon_{vel,75}$	Learned 75%-quantile velocity threshold for EoR plausibility check
ε_{rest}	Minimum resting threshold for EoR plausibility check
ε_{static}	Minimum duration threshold of the static phase for EoR plausibility check
$\boldsymbol{\xi}$	Tolerance value of the upper and lower boundaries of the clusters, $\mathbb{R}^{2\times 1}$
$\boldsymbol{b}_{+}^{(\chi)}$	Upper boundary value of the cluster χ, $\mathbb{R}^{2\times 1}$
$\boldsymbol{b}_{-}^{(\chi)}$	Lower boundary value of the cluster χ, $\mathbb{R}^{2\times 1}$
ε_n	Increase of the a priori value of the focused AoI between the latest sample points n and $n+1$
$\Delta_n^{(\kappa)}$	Proportinate differrence of an a priori value of the AoI κ
t_{blink}	Duration of an eye blink in ms
th_{min}	Minimum duration threshold of an eye blink
th_{max}	Maximum duration threshold of an eye blink
tol	Tolerance of the eye blink detection
m_{word}	Size of an encoded word
$H(X)$	Entropy of the variable X
γ	Relevance threshold of the information gain
m_{seq}	Sequence size of the moving window of the driver-activity approach
n_{seq}	Step size of the moving window of the driver-activity approach
Σ_{SAX}	Alphabet of SAX symbols
$\hat{q}_n^{(Q)}$	Estimation of the Q-th quantile at the n-th sample
$\Delta_{ref}^{(Q)}$	A priori learned distance of the Q-th quantile to the median
$q_{ref}^{(Q)}$	Averaged a priori learned quantile

Abbreviations

ACC	**A**daptive **C**ruise **C**ontrol
ADAS	**A**dvanced **D**river **A**ssistance **S**ystem
AoI	**A**rea-**o**f-**I**nterest

BASt	**B**undes**A**nstalt für **St**raßenwesen
BMM	**B**ayesian **M**ixture **M**odel
CAD	**C**omputer-**A**ided **D**esign
CAN	**C**ontroller **A**rea **N**etwork
CCD	**C**harge **C**oupled **D**evice
CMOS	**C**omplementary **M**etal-**Ox**ide-**Semiconductor**
DAR	**D**river-**A**ctivity **R**ecognition
DCS	**D**river **C**ontrollability **S**et
DTR+Q	**DisTR**onic Plus with Steering Assist
EOG	**E**lectro**O**culo**G**ram
EoR	**E**yes-**o**n-**R**oad
GMM	**G**aussian **M**ixture **M**odel
GUI	**G**raphical **U**ser **I**nterface
HAR	**H**uman **A**ctivity **R**ecognition
HiL	**H**ardware **i**n the **l**oop
I-AOI	**I**dentification **A**rea-**o**f-**I**nterest
I-DT	**I**dentification **D**ispersion-**T**hreshold
I-HMM	**I**dentification **H**idden **M**arkov **M**odel
I-KF	**I**dentification **K**alman-**F**ilter
I-MST	**I**dentification **M**inimum **S**panning **T**ree
I-VT	**I**dentification **V**elocity-**T**hreshold
IR	**I**nfra**R**ed
KNN	**k**-**n**earest **n**eighbors algorithm
Ko-HAF	**Ko**operatives-**H**och**A**utomatisiertes **F**ahren
LED	**L**ight **E**mitting **D**iode
MABX	**M**icro**A**uto**B**o**X**
MERCY	**M**oving **E**stimation **C**lassification
NEBAF	**NE**bentätigkeiten **b**eim **A**utomatisierten **F**ahren
NHTSA	**N**ational **H**ighway **T**raffic **S**afety **A**dministration
NIR	**N**ear-**I**nfra**R**ed
RBF	**Radial Basis Function** Kernel
RCP	**R**apid **C**ontrol **P**rototyping
SAE	**S**ociety of **A**utomotive **E**ngineers
SVM	**S**upport **V**ector **M**achine
VMP	**V**ariational **M**essage **P**assing
VOG	**V**ideo-**O**culo**G**raphy

1 Introduction

Worldwide nearly 91 million vehicles, including automobiles and commercial vehicles, were produced in 2015, an increase of about 1.1% compared to [1]. Furthermore, the total number of vehicles used worldwide in 2014 exceeded 1.2 billion units, corresponding to an increase of 38% compared to the number in 2005 [2]. This steady growth of vehicles on the roads, especially in developing countries such as China or India, comes with many problems such as traffic jams, a significant increase in carbon emission, and an increasing number of accidents. At the same time, Figure 1.1 shows that the trend of a decreasing number of accidents in developed countries such as Germany, seems to be diminishing or even starting to reverse itself [3]. One of the reasons for this behavior is that the current active and passive advanced driver assistant systems (ADAS) are beginning to reach their limits in terms of accident avoidance and mitigation. As a result, many researchers around the world are searching for new solutions to challenges posed by the rising number of vehicles.

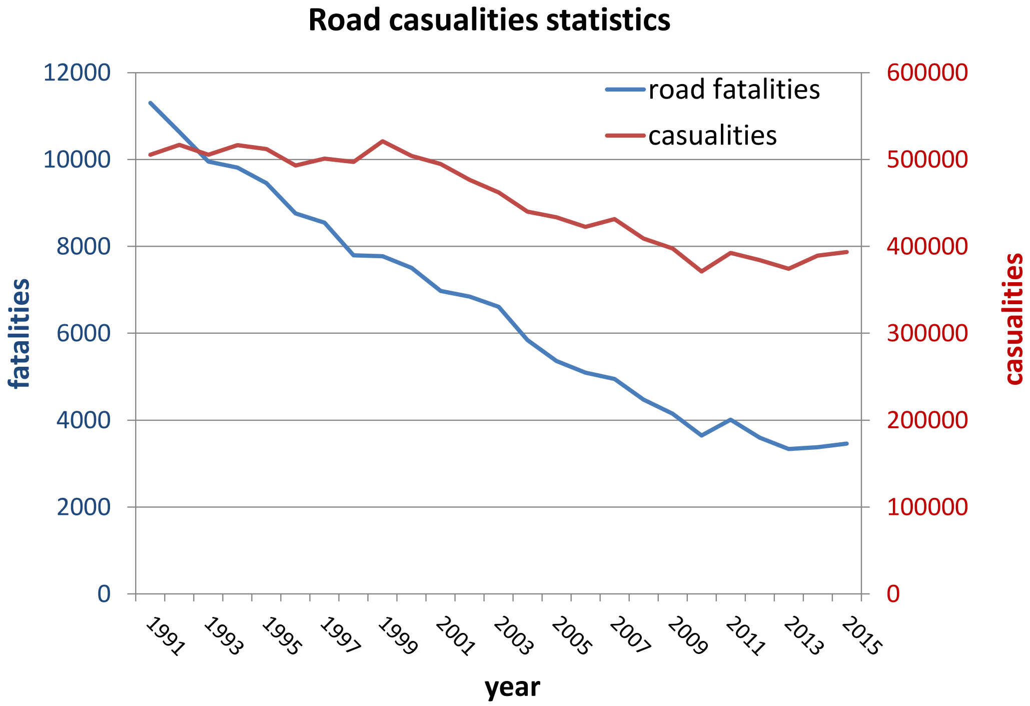


Figure 1.1: Total number of road fatalities and causalities in Germany over the last 24 years.

Currently, the greatest hope for confronting all the mentioned challenges and reaching the goal of accident-free driving is being placed in the concept of *automated driving*. The most common causes of accidents or traffic jams are still mistakes made by human drivers [4]. By taking the driver out-of-the-loop, i.e. by transferring the control of the vehicle to an automated driving function, mistakes by the driver can be prevented. As a consequence, the driver's comfort and safety will be enhanced, especially for long-term road trips in monotonic environments. The next step towards the fully automated vehicle is the level of conditional automation, where the automated driving function will take over the lateral and longitudinal driving task as well as the responsibility for the vehicle for a limited amount of time. Moreover, the driver will no longer be required to monitor the road or the automated driving function, which is still necessary for today's ADAS. Instead, the driver will be able to perform secondary tasks, such as reading news, watching a movie, or just relaxing in the driver seat. Which secondary tasks will be legalized in such automated vehicles, however, is still an open topic. If the system reaches its limitations, e.g. no more lane markings are visible, it will warn the driver to take over the driving task again after a specified time interval.

At this point the question arises as to how much take-over time is adequate. To answer this question, many aspects need to be considered. First of all, the maximum available take-over time is limited by the visual range of the applied sensors. The lower their range, the shorter the take-over time at sustained speeds. The range of the sensors depends on many factors, such as the geometry of the road or the weather conditions. As a consequence, there will most likely not be one fixed take-over time, but rather a time interval from the minimum to the maximum take-over time. Multiple studies are currently being conducted by different research institutes and automobile companies to determine appropriate take-over times and to ensure the safe transition from automated driving to the human driver [5], [6], [7]. However, the take-over quality of the driver is subject to considerable inter- and intra-individual variations. It was shown that it is influenced, among other things, by the complexity of the traffic situation [8], the performed secondary task [9], [10], or the gaze behavior of the driver [11]. That is why it may not be possible to select a single take-over time adequate for all situations and each driver. To increase the take-over time for such systems, better sensors or lower velocities need to be considered, and that is usually an expensive and inacceptable option.

Instead of adapting the requirements for using such a conditionally automated function, e.g. by decreasing the allowed velocity or by prohibiting some secondary tasks, systems for driver monitoring could be used to automatically detect drivers unable to take-over in time. To do so, the previously mentioned impact factors such as the gaze behavior of the driver or the performed secondary task need to be recognized online in the vehicle and applied for a take-over classification. If such an ADAS predicts a low take-over readiness of the driver during a conditionally automated drive, further measures need to be performed to ensure a safe and comfortable transition. For example, the driver may be asked to perform gazes at the road for reorientation and getting the driver back into the loop. If the driver ignores this warning, the amount of allowed secondary tasks could be reduced to tasks which are less distracting. On the other hand, if the driver follows the guideline or if the ADAS predicts a high take-over readiness, no further measures need to be taken. Additionally, the number of secondary tasks could even be expanded temporarily. Such an expansion of the allowed secondary tasks would be a relevant reason to buy a vehicle with conditionally automated driving functions. Vehicle manufacturers will therefore most likely compete in the future to enable as many secondary tasks as possible.

1.1 Scope and Contribution of this Thesis

The main contribution of this work is the first approach to a novel ADAS for assessing the driver's take-over readiness in conditionally automated driving scenarios. The automated classification is based primarily on features extracted from the visual search behavior based on an in-vehicle driver monitoring system. Moreover, the work highlights the necessity of automated approaches for detecting gazes at the road and driver-activity recognition for the feature extraction, since current state-of-the-art methods face multiple challenges such as: i) only a coarse separation of driver activities possible, ii) necessity for costly and time-consuming calibrations, and iii) no adaption to conditionally automated driving scenarios. Hence, novel approaches for detecting gazes at the road and recognizing the driver's activity will be developed, implemented, and evaluated in this work. Further, this work focuses on analyzing the application of these approaches under realistic conditions in the vehicle. Thus, the evaluation of the driver assistance system and its various subsystems is based on the data recorded during thorough driving simulator and real driving studies with reference as well as close-to-production measurement systems. Since a robust detection of basic eye movements is crucial for some of the proposed methods for driver-activity recognition and Eyes-on-Road detection, a novel eye movement classification method especially designed for the automated environment is introduced. This is a particularly challenging task for conditionally automated driving scenarios due to the highly dynamic environment, inter- and intra-individual differences, varying lighting conditions, and other issues. Moreover, the task-individual and highly varying eye movement behavior is proofed in conditionally automated driving scenarios by means of the introduced classification method for basic eye movements. All these components are incorporated into the final ADAS validated at the end of the work.

1.2 Organization of this Thesis

The introduction and motivation in Chapter 1 is followed by some necessary basic knowledge about the different levels of automation, the structure of the human eye, and head- as well as eye-tracking approaches in Chapter 2. Furthermore, all of the conducted driving studies as well as an approach for generating a gaze direction based on intrusive eye- and head-tracking systems are described at the end of this chapter. In Chapter 3, a novel eye movement classification method especially designed for the automated environment is introduced. This approach is applied to proof the task-individual and highly varying eye movement behavior in conditionally automated driving scenarios. The findings were published on the symposium on Eye Tracking Research and Applications [12]. Chapter 4 covers approaches for detecting Eyes-on-Road and possible fallback strategies in case of missing or poor gaze estimation which are contained in the published patent [13] and a paper of the IEEE Intelligent Vehicle Symposium [14]. Chapter 5 is devoted to driver-activity recognition including two novel approaches proposed in the paper for the IEEE Conference on Intelligent Transportation Systems [15], on the 15th International Stuttgarter Symposium [16] and as contribution to the IEEE Transactions on Intelligent Transportation Systems Journal [17]. The chapter concludes with the description of the necessary adjustments for transferring approaches for driver-activity recognition to an online-setting. Finally, the results of the previous chapters are combined in Chapter 6 for a proof of concept of the proposed ADAS. This prototype is validated based on take-over situations of a driving simulator study which were originally published as contribution in the IEEE Transactions on Intelligent Transportation Systems Journal [18]. The work concludes with Chapter 7 where the results are summarized and potential for future developments is outlined.

2 Fundamentals

2.1 Levels of Automation

A common mistake in public discourse on automated driving is that the terms *automated*, *autonomous*, *highly automated*, and *fully automated* are erroneously used as synonyms. However, it is necessary to differentiate between these terms and the different levels of automation. There are various taxonomies of the automation levels concerning on-road vehicles, e.g. the categorization of the BASt[1] in [19] or of the NHTSA[2] in [20]. However, this work applies the taxonomy defined in [21] by the SAE[3], since it is the most common one used in the automotive industry. The different levels of automation are distinguished by reference to the driving tasks taken over by the automated driving function and the behavior of the vehicle in traffic situations for which this function was not designed. The automation levels and a short explanation are given in Table 2.1.

The number of vehicles without any supporting driver assistance systems usually decreases annually. In general, off-the-shelf vehicles are equipped with cruise control or even adaptive cruise control (ACC) systems to take over the longitudinal control of the vehicle. Vehicles with partially automated driving functions, usually distributed under other names, have been available for series vehicles for many years. For example, in 2013 Mercedes-Benz introduced *Distronic Plus with Steering Assist*, an ADAS for lateral and longitudinal control of the vehicle [22]. Today, similar systems are available for vehicles of many other automobile manufacturers, e.g. [23], [24]. An increased risk is given by vehicles with partially automated driving functions without obvious prompts for drivers not paying attention to the traffic environment or to the automated driving function, e.g. not keeping their hands on the wheel. While traditional automotive companies as Daimler AG or Audi AG realize a strict hands-off detection which warns the driver and finally turns off the automated driving function for inattentive drivers in a minimum amount of time, startup companies such as Tesla Motors perform warnings less frequently. Due to the combination of fewer warnings and automated driving functions with increased performance and availability, the impression is created that the vehicle is already driving in a conditionally automated setting. This may lead to fatal consequences [25].

At the moment, there are no conditionally automated driving functions available for series vehicles. Table 2.1 indicates the major challenge of the development of current assistance systems by means of the double dividing line, namely the development step from the

[1] Bundesanstalt für Straßenwesen

[2] National Highway Traffic Safety Administration

[3] Society of Automotive Engineers

partial to the conditional automation level. Up through the degree of partial automation, the driver holds the responsibility for the vehicle at all times; the automated driving function takes over the responsibility in the event of conditionally automated driving scenarios in specified traffic scenarios. This would be the first time that an automated driving function takes over the responsibility for the vehicle. Hence, the driver is able to perform secondary tasks while driving. However, which secondary tasks will be legalized in the future has not yet been decided. Note that conditionally automated driving functions will experience situations which they can no longer handle, e.g. if no more lane markings are visible. In such situations, the driver needs to take-over the control and responsibility of the vehicle again. These situations are called take-over situations. Since drivers have only a limited amount of time to take over the control of the vehicle, they are rushed and experience a stress situation.

Take-over situations reflect the paradox of automation described by Bainbridge in [26]. Although automation is designed for supporting the human operator and facilitating the actual task, especially for monotonous and long-term tasks, the human operator is assigned to the novel role as a monitor, which is even less suited to humans than the original tasks. Many examples of catastrophic failures can be provided by civil aviation as listed in [27], where the role of a pilot is already defined as a monitor and fallback level. Such failures can also be assumed to occur for conditionally automated driving scenarios on the road and, therefore, must be taken into account in designing future automated driving functions. Based on the level of conditional automation, higher levels of automation will emerge, guaranteeing an increased performance and system availability in a rising number of traffic situations. This development will finally culminate in the fully automated vehicle, which transports the passengers to a defined destination without any need of intervention by the human driver. The term *autonomous* usually describes systems equal to the level of full automation, but without any passengers at all. This would be of interest for use-cases such as *robot taxis* or *autonomous parking garages*. The following work focuses on systems at the level of conditionally automated driving, especially on the system degradation occurring at levels with no automation in take-over situations.

Name	Description	Driving Task	Secondary Task	Fallback Level	System capability
No Automation	• Driver performs all aspects of the driving task without any automation • Warnings in demanding situations are possible	None	None	Driver	n/a
Driver Assistance	• System takes over either lateral or longitudinal control • Driver executes remaining tasks	Lateral or Longitudinal	None	Driver	Some
Partial Automation	• System performs longitudinal and lateral control • Driver monitors continuously the system and traffic environment • Driver can take over immediately	Both	None	Driver	Some
Conditional Automation	• System performs all driving tasks • Driver does not need to monitor the system or the environment • Driver has to take over within a specified interval	Both	Some	Driver	Some
High Automation	• System can restore a minimal risk condition in take-over situations	Both	Some	System	Some
Full Automation	• System is available unconditionally at all times • Driver may act as a manager of the system	Both	All	System	All

Table 2.1: The summary of the different levels of automation and their corresponding attributes taken from the taxonomy of the SAE [21].

2.2 The Human Eye and its Movements

The human eye is the main sensory organ and can be seen as extended components of the human brain. Hence, the eyes are often used for inference to cognitive processes or to predict the subject's mental state, e.g. being drowsy or attentive. In this study, eye movements of drivers engaged in different secondary tasks are used to derive features for recognizing the currently performed activity. Therefore, the eye movements need to be recorded, detected, and classified by means of appropriate measurement systems and algorithms. In this section, the structure of the human eye as well as the basic eye movements will be introduced to support the understanding of the later presented measurement systems and algorithms. Since the human eye is such a sophisticated and well-studied organ, there is an extensive amount of literature available, describing each aspect of the eye in detail. A coarse overview over the parts of the human eye relevant for the later algorithms and systems will be included here. For those interested in a more detailed description of this topic, please refer to [28] and [29].

2.2.1 Structure of the Human Eye

The eye is a nearly spherical organ with a radius of about 1.2 cm, except for the small bulge at the front, and is protected by the orbit and the upper and lower eyelid. A profile of an eye is shown in Figure 2.1, including the labeled relevant parts for this section. The visible, frontal part of the human eye includes the iris, the pupil, the sclera, the limbus and the cornea. The cornea forms the translucent layer at the frontal part of the eye and is responsible for a large portion of the light refraction. Behind the cornea, the pupil can be seen as a dark, circular opening surrounded by the dyed iris. The pupil enables light to enter the inner of the eye and, therefore, appears dark. Depending on the lighting conditions, the size of the pupil can be adapted by means of the musculature of the iris to control the amount of light entering the eye. The limbus represents the border between the iris and the surrounding white sclera, which is a further protection of the eye. Directly behind the pupil is a biconvex lens. The lens focuses all the light entering the pupil into a collimated ray on the retina located at the back of the eye. Since the retina is a light-sensitive layer on account of its photoreceptor cells, an image in the form of electrical impulses is sent over the optic nerve to the brain. There are two light-sensitive photoreceptor cells, namely the rods and cones. While rod cells are responsible for night vision, cone cells enable the human eyes to perceive colors. Depending on the literature, the total number of cone and rod cells varies between 90 million and 120 million rod cells and between 4.5 million and 6 million cone cells [30] [31]. However, the ratio of $1/20$ between the two types of photoreceptor cells remains the same for both reported numbers. These cells, provided with nourishment by the vascular choroid layer, are not uniformly distributed over the retina. Most of the cone cells are located at the fovea centralis, generating the area of sharpest vision. The fovea centralis is subject to significant individual differences and may differ up to 5° from the optical axis[4] [32]. Rod cells on the other hand, are not found at the fovea

[4] Optical axis describes the line passing through the center of the cornea, lens, and imaginary pivot of the eye.

centralis, but at the outer edges of the retina. Hence, rod cells are responsible for the peripheral vision. Another interesting location concerning the photoreceptor cells is the blind spot. At this point, neither rod nor cone cells exist and, therefore, no visual information can be perceived. The blind spot is located on the optic disc area where the optic nerve, also called Carnial nerve II, exits the eye. The visual information is transported via the optic nerve to the corresponding areas of the brain for processing visual information.

The human eye is able to rotate about all three-dimensional axes by means of the extraocular muscles attached on the upper (M. rectus superior, M. obliquus superior), lower (M. rectus inferior, M. obliquus inferior), and the lateral (M. rectus lateralis, M. rectus medialis) side of the eyeball.

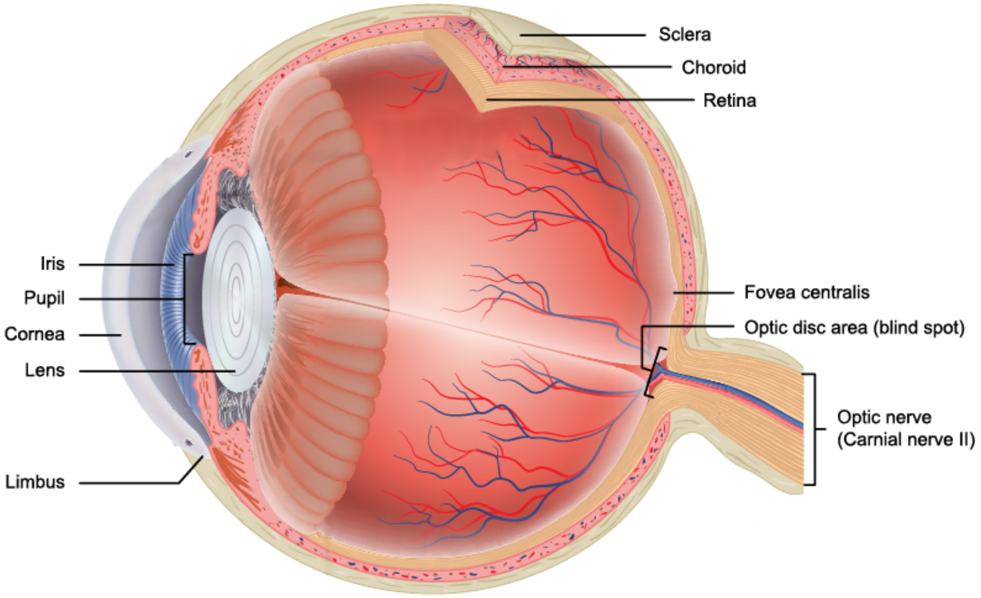


Figure 2.1: Anatomy of the human eye adapted from [33].

2.2.2 Movements of the Human Eye

According to Leigh and Zee [34], there are six basic eye movements in daily life, namely *saccades*, *fixations*, *smooth pursuits*, *vergence*, *vestibulo-ocular reflex*, and *optokinetic reflex*. For the majority of our awake-time, our eyes remain motionless in a temporal position and perform *fixations*, i.e. focusing on a given visual target for perceiving visual information. In fact, the eye performs tiny movements even during fixations, so called micro-saccades, helping the eye to gather enough stimuli for the photoreceptor cells to keep the image sharp. It is usually assumed that the subject is paying visual attention to the focused target, although some rare phenomena such as *Looked-but-failed-to-see* may result in the missing perception of crucial information occurring in the field of view. According to [35], the average duration of a fixation is about 200 ms to 300 ms. However, Schweigert found that the duration decreases to 80 ms - 100 ms for fixations performed when driving

manually [36]. In case of shorter fixations, no visual information can be perceived.
By means of rapid eye movements called *saccades*, the eyes change from one visual target to another. During a saccade no visual information can be gathered and that is why saccades are often performed simultaneously with necessary eye blinks. This is an evolutionary mechanism, since the human eye has only a small area of ±1° of highly sharp vision, called the *foveal area*, corresponding to the image on the fovea centralis. To scan wide areas as fast as possible, the eye reaches velocities of up to 1000 °/s during saccades. Saccades are characterized by their amplitude, direction, and duration. If the eye fixates a moving object, e.g. focusing on the ball while watching a tennis match, it performs particularly smooth movements. Hence, these basic eye movements are called *smooth pursuits*. The velocity of smooth pursuits ranges between 15 °/s and up to 60 °/s at maximum [35], [37]. This velocity profile enables the eye to maintain a sharp image of the moving object. *Vergence* describes an eye movement type in which the eyes rotate simultaneously in an opposite direction about a vertical axis. By means of this eye movement type, *diplopia*[5] can be prevented. The *vestibulo-ocular reflex* is a compensational eye movement in case of gaze shifts with subsequent head movements. For example, consider a driver checking the right exterior mirror before changing lanes. Since the visual angle is so large that spanning this distance solely with the eyes feels uncomfortable or is not even enough to see the right mirror, the head starts moving to the right until an adequate position is reached. In this case, the eye reaches the visual target first. To already perceive information while the head is still moving, the eyes need to move for the same amount in the opposite direction. The last eye movement is the *optokinetic reflex*, which is a combination of a saccade and a smooth pursuit. This eye movement type can be monitored when the eyes follow a moving object, i.e. performing a smooth pursuit while the head remains motionless, e.g. when sitting in a train and observing an object through the window. As soon as the object moves out of the field of view, the eye jumps back to the starting position of the smooth pursuit, i.e. the eye performs a saccade. This study analyzes some of these basic eye movements, namely saccades and fixations, to detect and classify the secondary task and the gazes on the road performed by the driver during conditionally automated driving scenarios.

2.3 Measurement Methods

2.3.1 Head-Tracking

Head-tracking is used to determine the subject's head movements including the position and rotation angles and is being considered for numerous applications such as future user-interfaces for disabled persons or applications in the gaming or military sector. Head-tracking sensors comprise intrusive as well as non-intrusive measurement variants. Intrusive methods include accelerometers, gyroscopes, or laser-based approaches (see 2.3.2). Non-intrusive methods with practical relevance can be limited to optical tracking approaches with one or multiple cameras. Since an intrusive system is described in detail in Section 2.3.2, this section focuses on optical head-tracking applied by remote camera

[5] Perceiving the same object at two different locations.

systems as described in Section 2.3.5.

The first step of an optical head-tracking system always contains the detection of the driver's head and face. This is usually done by applying the *Viola-Jones algorithm* [38], which is able to detect objects, such as faces, in grey-scaled images by means of rectangular Haar features. In case of a stereo camera system, the information about the disparity may also be used to locate the driver's body and to remove the background of the image followed by the separation of the head and torso [39]. This separation can be performed by means of a statistical model which parameters can be trained by Bayesian or maximum likelihood estimators. If the driver's face is detected, it is used as the initial state for the actual model. According to [40], there are three major categories of models used for detecting facial landmarks such as the corners of the eyes or mouth, and estimating the head pose: the *Template-based Models*, *Active Shape Model*, and the *Active Appearances Models*.

Template-based Models

In general, template-based methods in computer vision try to correlate desired aspects with different areas of the recorded image and determine the maximum correlation between these aspects and areas. For head pose estimators, template-based models compare facial features of an unseen image with the facial features of labelled images of an a priori acquired training set, called templates, and searches for the best fitting one. The search and fitting process can be performed efficiently and with similar performance to known Active Appearance models [41], due to the combination of texture and shape models by the nearest neighbour algorithm. Clearly, an increasing training set and number of templates improves the estimation. However, recording and labeling of such images is a time-consuming task. Instead, head morphing approaches are used to generate artificially a hugh number of images with the corresponding Ground Truth of the head pose. Nevertheless, to handle the enormous variations of subjects and head poses, such methods suffer from CPU-intensive algorithms and a high memory consumption. Note that similar methods exist for eye-tracking approaches (see 2.3.3) which show the same disadvantages as the methods for head pose estimation and, additionally, cannot reach an accuracy of better than 5°.

Active Shape Models

To detect particular non-rigid objects, e.g., in biomedical images or in recorded human faces, Cootes et al. [42] introduced the active shape models. Based on a training dataset of preferably many varying faces, common feature points are determined to establish an initial shape model, called Point Distribution Model. This statistical model describes the allowed variations of the shape model of the particular object class. The structure of the object is described by a defined number of connected points and can be adapted to the object shape by means of iterative algorithms. The larger the number of points and enclosed areas, the more accurate the shape model. In comparison to Active Contour Models proposed by Kass et al. [43], Active Shape Models proved to be robust with regard to object location tasks due to the constraining to maintain a similar shape as the training set. However, these models may not converge to appropriate solutions and are not able to cover variations not part of the training set.

Active Appearance Models

In addition to the above mentioned points, Active Shape Models show one major point of criticism: They do not incorporate any information about the texture of the object. This would further improve the robustness of the models. Hence, Cootes et al. introduced Active Appearance Models in [44], an extended version of the Active Shape Models. These models consider the gray-level information of the enclosed areas of the objects. The correlation between errors of the texture models and the model parameters are learned to decrease the processing time of the matching by means of an iterative matching algorithm. Cootes et al. showed that this enhancement of the Active Shape Models converge rapidly and reliably given an appropriate start point. Nowadays, Active Appearance Models are the method of choice with regard to head pose estimation applications.

2.3.2 laserBird

For highly accurate detection of the head position and rotation, this study used the laser scanner *laserBird* developed by Ascension[6] [45]. The *laserBird* consists of a scanner and a head-mounted sensor as shown in Figure 2.2. The scanner itself emits the fan-like laser beams with a wavelength of 785 nm detected by the three sensors of the head-mounted part, called the *bird*. Measuring the laser beams with the three photodiodes enables the calculation of the stretched plane and of the corresponding center of mass and, therefore, the calculation of the position and rotation of the bird. The *laserBird* measures with a sampling rate of 240 Hz and writes this data on a RS232 interface. The accuracy of the head position and rotation is 0.7 mm in x-, y-, and z-direction and 0.5° about each axis in the three dimensional space according to the supplier. However, these accuracies are only valid as long as the detectors of the bird do not exceed an orientation range of ±85° with respect to the scanner and the bird is in a given region, the Performance Motion Box, defined by the interval distances

$$[pmb_x] = [+43, +99] \quad [\text{cm}] \tag{2.1}$$

$$[pmb_y] = [-23, +23] \quad [\text{cm}] \tag{2.2}$$

$$[pmb_z] = [-30, +30] \quad [\text{cm}] \tag{2.3}$$

between the head-mounted part and the scanner. To use the *laserBird* inside of a vehicle, the scanner has to be mounted above the front passenger seat so the head can be seen without any blockage. The mounting position of the scanner, which at the same time is the origin of the *laserBird* coordinate system, was measured in a Mercedes-Benz E-class (W212) with high accuracy at the start-up factory of the Daimler AG in Sindelfingen and

[6] http://www.ascension-tech.com/

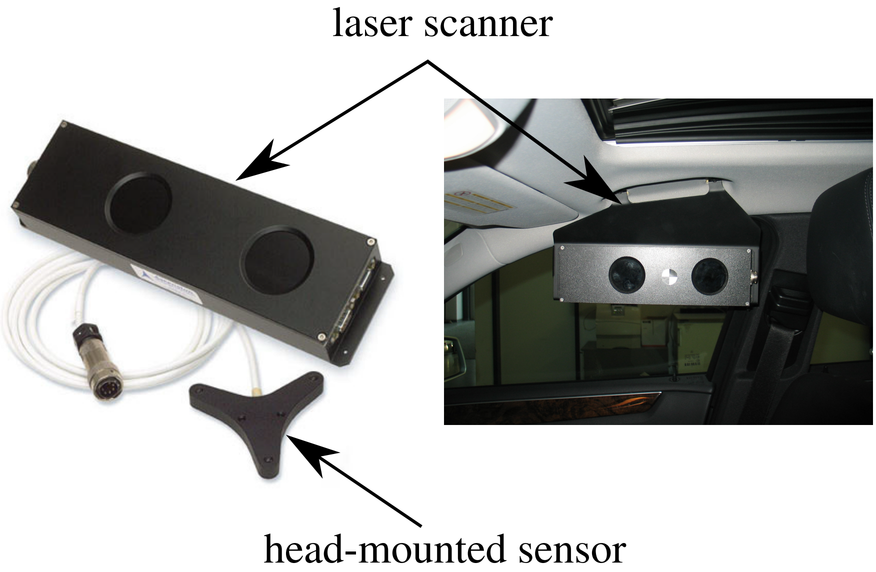


Figure 2.2: On the left side, the two main components of the laserBird, the head-mounted sensor and the laser scanner, are shown. On the right side, the mounting position of the laserBird inside the vehicle is shown.

is given by the vector $\mathbf{lb}_{init}$. This mounting position of the scanner assures that the head of the driver is within the defined Performance Motion Box. The 3-dimensional head position

$$\mathbf{lb}_{pos,in} = \begin{pmatrix} lb_{pos_x,in} \\ lb_{pos_y,in} \\ lb_{pos_z,in} \end{pmatrix} \quad \text{[inches]} \tag{2.4}$$

with

$$\forall t \in \{x,y,z\} : lb_{pos_t,in} \in [6'', 72''] \subset \mathbb{R} \tag{2.5}$$

is measured in inches and the head rotation

$$\mathbf{lb}_{rot,deg} = \begin{pmatrix} \phi_{lb,deg} \\ \theta_{lb,deg} \\ \psi_{lb,deg} \end{pmatrix} \quad [°] \tag{2.6}$$

with

$$\phi_{lb,deg} \in [-180°, 180°] \subset \mathbb{R} \tag{2.7}$$

$$\theta_{lb,deg} \in [-85°, 85°] \quad \subset \mathbb{R} \tag{2.8}$$

$$\psi_{lb,deg} \in [-85°, 85°] \quad \subset \mathbb{R} \tag{2.9}$$

is given as a 3-dimensional vector in degrees. To unify the subsequent calculations and units, each position value is transformed from inches into millimeters by

$$\mathbf{lb}_{pos} = \begin{pmatrix} lb_{pos_x} \\ lb_{pos_y} \\ lb_{pos_z} \end{pmatrix} == \begin{pmatrix} lb_{pos_x,in} \\ lb_{pos_y,in} \\ lb_{pos_z,in} \end{pmatrix} \cdot 25.4 \quad [\text{mm}] \tag{2.10}$$

and each rotation value from degrees to radians by

$$\mathbf{lb}_{rot} = \begin{pmatrix} \phi_{lb} \\ \theta_{lb} \\ \psi_{lb} \end{pmatrix} == \begin{pmatrix} \phi_{lb,deg} \\ \theta_{lb,deg} \\ \psi_{lb,deg} \end{pmatrix} \cdot \frac{\pi}{180} \quad [\text{rad}]. \tag{2.11}$$

2.3.3 Eye-Tracking

Eye-tracking describes a set of methods for determining eye movements and sometimes the subject's gaze direction. There are various methods available for this purpose such as the Electrooculogram (EOG) [46], search coil contact lenses [47] or video-oculography (VOG) [48]. In the following, the video-based approach will be described in detail since it will be used for recording the driver's eye movements in this study. An eye tracker usually consists of one or more lighting sources, typically infrared (IR) or near-infrared (NIR) light, and a camera which is focused on the eyes of the subject. The location and method-specific features of the frontal eye and of the reflections of the infrared illumination are extracted of the recorded image by means of computer vision algorithms. Based on geometrical correlation of the features, the gaze direction can be calculated. Basically, there are two types of tracking approaches for VOG. If the limbus and the center of the iris are tracked for further calculations, it is referred to as *limbus-tracking*. However, for most subjects this approach suffers from a coverage of the vertical edges of the iris, which makes this an unsuitable method for measuring vertical eye movements. Instead, the transition from the pupil to the iris is usually used to track the center of the pupil. This approach is called *pupil-tracking*. Depending on the illumination setup, the pupils appear bright or dark in the recorded image. If the illumination source is located near to the optical axis of the camera, the pupils seem to light up due to the reflected light of the retina. This bright pupil approach performs well independent of the iris color of the subject, but is not suitable for outdoor scenarios with varying lighting conditions. For a more stable tracking performance in outdoor applications, the dark pupil approach is applied. Therefore, the illumination source is positioned away from the optical axis of the camera, so that no light will be reflected of the retina into the direction of the camera.

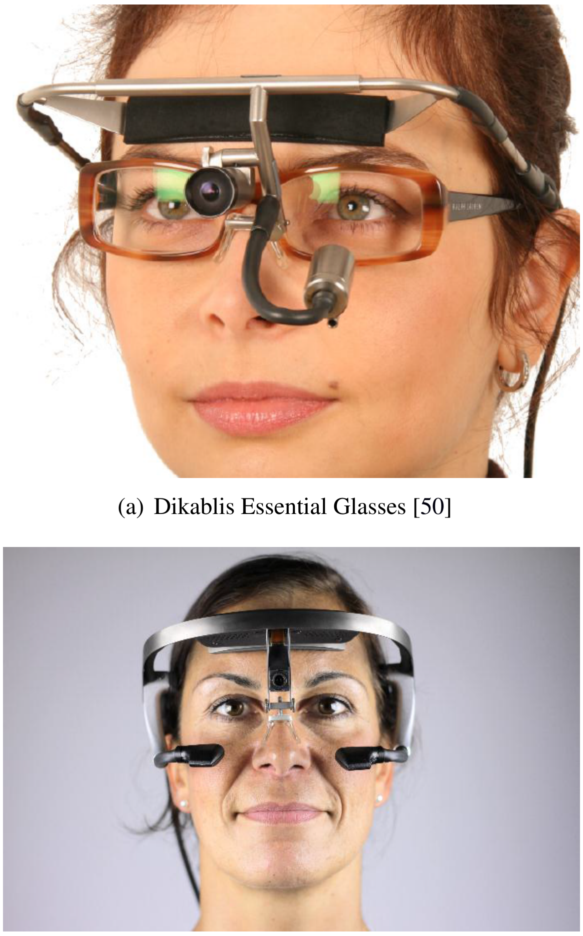
The most common approach to determine the gaze direction is called the *pupil-corneal reflection approach* [48]. It is assumed that one camera and two illumination sources are available. Besides the center of the pupil, the center of the reflections on the cornea of the illumination sources are identified on the recorded image. These reflections are often called *glints*. A vector between the glints and the pupil center is constructed to determine the gaze direction. However, the glints, the pupil center, and the gaze direction are given in image coordinates. To transform these points into world coordinates, a geometric model of the eye is constructed. The parameters of the model have to be estimated by means of an individual calibration including fixations on usually nine calibration points. To reduce the number of calibration points to only one point, multiple cameras and illumination sources have to be applied simultaneously [49]. To completely avoid any individual calibration, a universal geometric model of the eye has to be applied. However, such universal models cannot compensate for the individual variance of the location of the fovea centralis. Hence, the accuracy of this approach is limited to the individual variance of the fovea centralis of 5°. On the other hand, an estimation of the gaze direction for just one illumination source is only possible for a completely stationary head.

2.3.4 Dikablis

In this work, two different versions of the head-mounted eye-tracking system *Dikablis*, manufactured by *Ergoneers*[7], were applied: *Dikablis Essential Glasses* [50] and *Dikablis Professional Glasses* [51]. The Dikablis Essential Glasses system represents the basic and simpler version of the two eye-tracking systems mentioned above. It consists of a spectacle frame with a mounted 384×288 pixel camera for recording the eye movements with a sample rate of 25 Hz (see Figure 2.3(a)). An illumination source with a wavelength of 875 nm is mounted beside the eye camera. The camera needs to be adjusted to the subject's face and eye structure by means of the flexible swan-neck mounting before starting the eye-tracking. Further, a field camera with a resolution of $768x576$ pixel is mounted between the eyes so that the current scene experienced by the driver can be monitored. Both camera images can be merged by means of a four point calibration to get an overlayed image of the scene and the corresponding gaze direction. The essential version of the Dikablis glasses is robust and easy to put on, even when mounting it on the head of another person. Compared to most other head-mounted eye-tracking systems it has the great benefit that it can be worn by subjects with their glasses on. This is an important requirement since a non-negligible share of the later test subjects drive while wearing glasses. Furthermore, it can be applied in driving scenarios without limiting the field of view of the driver since the eye camera can be positioned below the recorded eye. It can be worn simultaneously with the *laserBird* head sensor without any constraints.

The professional version of the Dikablis eye-tracking system differs from the essential version primarily in the quality of the recorded camera images. Dikablis Professional Glasses is a dual camera system consisting of two 384×288 px cameras for recording the eyes with an increased sampling rate of 60 Hz (see Figure 2.3(b)). Further, the resolution of the field camera is increased to $1920x1080$ px compared to the Dikablis Essential Glasses, resulting in an HD-ready image of the surroundings. However, these improvements of the sample rate and the camera resolution come with a decreased stability of the eye tracker and a swan-neck mounting which is harder to adjust to the different subjects. Especially the combination of the Dikablis Professional Glasses and the *laserBird* head tracker is uncomfortable to wear or not applicable at all for some subjects. Hence, there is the need

[7] http://www.ergoneers.com/



(a) Dikablis Essential Glasses [50]

(b) Dikablis Professional Glasses [51]

Figure 2.3: Both head-mounted eye trackers used in the later described experiments.

for a construction which enables the simultaneous usage of the *laserBird* and Dikablis Professional Glasses.

2.3.5 Remote Driver Camera

VOG and optical head-tracking come with the benefit of being non-intrusive. That means that they are suited for applications in vehicles since the driver's field of view is not reduced and the driver does not notice the eye- and head-tracking at all. However, applying head- and eye-tracking in series vehicles presents many challenges to the hard- and software. Depending on the weather and geographical location, the hardware has to withstand extreme temperatures and still maintain its functionality and performance. For example, component temperatures of 80° in the instrument cluster are common if the vehicle is parked in direct sunlight during summer times. In addition, high temperatures generate thermal-dependent noise, which is particularly critical for eye-tracking applications. Often extremely challenging for the soft- and hardware are the varying lighting conditions in the vehicle environment. The lighting conditions contain (sun-)light from every possible direction on the driver's face or directly on the imager of the camera. Hence, reflections on glasses, contact lenses, or other vehicle components may have a negative impact on the performance of the algorithms. Furthermore, since sunlight also contains light of the wavelength which is used by the IR illumination sources of the eye-tracker, the corneal reflection can no longer be detected robustly. Besides the lighting conditions, accessories represent another common challenge in this field of study. Sunglasses may have a low infrared permeability which at some level will prevent any eye-tracking approaches. Glasses, especially with high diopter or varifocal lenses, influence the refraction of the infrared rays of the eye-tracking and need to be taken into account by the applied models [52]. All the mentioned optical aids as well as most other accessories such as hats, scarfs, or disposable respirators cover parts of the driver's face and often some of the used facial features. Despite these driver-independent topics, there are many challenges concerning the inter-individual variations. The camera system has to consider all kinds of face and eye structures such as European (Caucasian), Asian, or African face and eye structures. Varying seating positions, mounting tolerances of the camera system, blocked camera view by the steering wheel or the driver's hands, and limited mounting space in the vehicle further increase the list of topics to be concerned with. In general, the distance between camera systems in series vehicles and the driver's eye are significantly larger compared to head-mounted camera systems such as the Dikablis eye-tracker. For purposes of comparison, images of the driver's eye are given for both types of camera systems in Figure 2.4. Hence, the algorithms of the driver camera have to estimate the gaze direction based on images with a significantly lower resolution of the driver's eye. As a result, the accuracy of the detection of the center of the pupil and the corneal reflections decreases deteriorating the overall gaze direction. In summary, remote driver camera systems in series vehicles face many challenges which usually lead to less accurate and unstable gaze estimations. A question that needs to be answered in this study is: Is the gaze estimation of a first generation of camera systems for series vehicles sufficient to be applied for classifying the driver's take-over readiness?

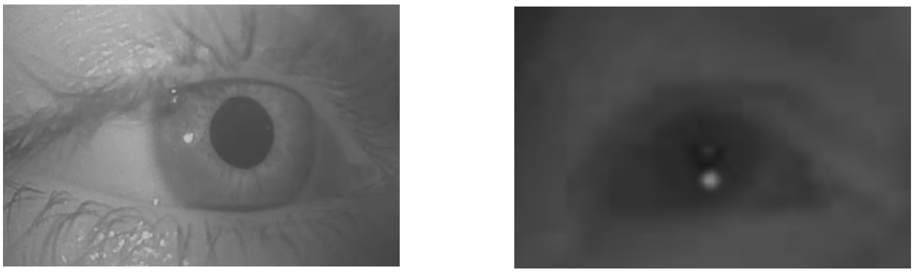


Figure 2.4: On the left side, a high resolution image of the head-mounted eye tracker Dikablis Professional is shown. On the right side is a low-resolution image of a near-to-production remote driver camera. The glint is visible in both images.

2.4 Experiments

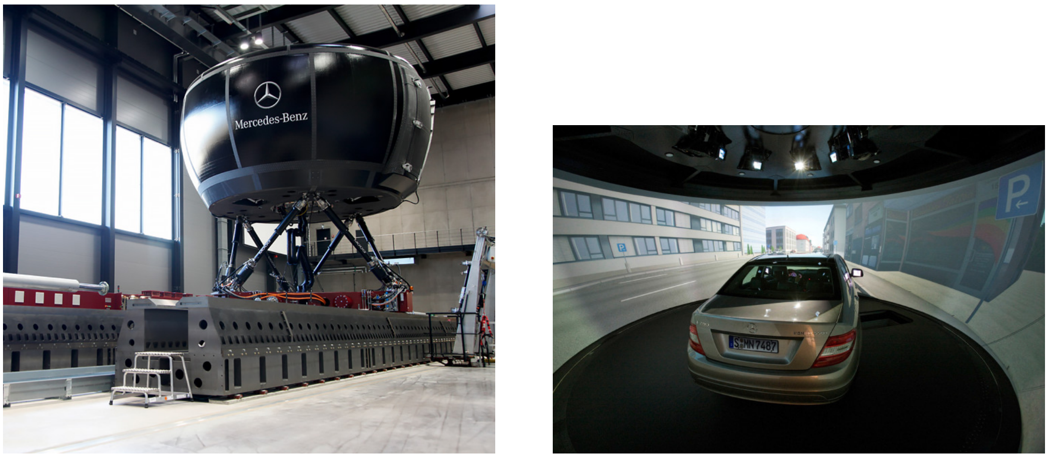
Different experiments have been conducted to evaluate the approaches proposed in this work. Since this work focuses on systems for an automated classification of the driver's take-over readiness in conditionally automated driving scenarios, two aspects are of particular interest for these experiments: i) driving in a conditionally automated setting and ii) creating take-over situations. Especially critical take-over situations can only be considered with justifiable expenditure by using a driving simulator. However, to provide investigations of the different algorithms for Eyes-on-Road detection and driver-activity recognition in the context of real-world driving data and close-to-production camera systems, a real driving study on a test track was performed as well. In this section, the applied simulator and the setup of the testing vehicle are described. Further, the procedure and scope of the driving simulator studies along with the real-world driving study will be presented.

2.4.1 The Mercedes-Benz Moving Base Simulator

One of the most sophisticated driving simulators in the world is in operation in the Driving Simulator Center of the Daimler AG in Sindelfingen, Germany [53]. This simulator is realized as a hexapod platform as shown in Figure 2.5(a) capable of performing accelerations in every possible direction. To realize high lateral or longitudinal accelerations, the hexapod is mounted on a 12 m carriage while the experimental vehicle can be positioned parallel or orthogonally to the carriage. This vehicle is located inside the dome on top of the hexapod and can be exchanged through an extra gate of the dome. That means that different vehicle models can be used for different experiments. The inside of the dome is shown in Figure 2.5(b). A 360° view as well as the presented content of the exterior and interior mirrors are simulated by means of the visual software Pixel Transit [54]. In addition to the realistic accelerations and vehicle environment, real engine and wind sounds can be simulated over the speakers in the vehicle cabin. The interior of the vehicle is equipped with multiple RGB cameras at different angles to monitor and record the driver's face, the footwell, the area of the steering wheel, and the center console.

2.4.2 Testing Vehicle

A Mercedes-Benz S-class (V222) with the Intelligent Drive extra equipment was selected as the test vehicle for the real-world driving study. This equipment includes the *Distronic Plus with Steering Assist* system mentioned in Section 2.1 which represents a partially automated driving function. However, this functionality was modified by deactivating the hands-off detection of the series vehicle system such that no warnings occurred and the driver was able to drive freehand. A tablet device with a 9.7″ display was mounted at the center console to enable the driver to perform secondary tasks. Further, the vehicle was equipped with a near-to-production camera system prototype provided by FOVIO. As shown in Figure 2.6, the stereo camera system was mounted on the steering column tube

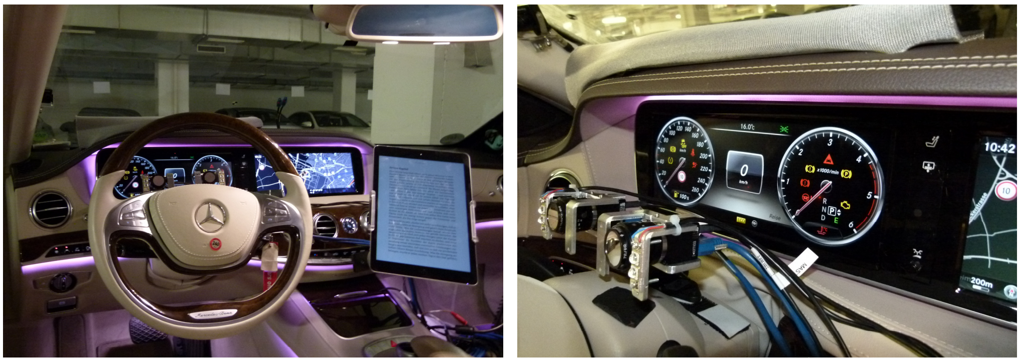


(a) Hexapod on carriage

(b) Vehicle inside of the dome

Figure 2.5: Mercedes-Benz Moving Base Simulator: external and inner view

in front of the driver. The illuminators are each mounted beside to the CMOS[8] camera sensors. The resolution of each camera is 1.3 MP (1280×1024) with a sample frequency of 60 Hz. The camera system was connected to the vehicle's private CAN bus and provided among other things the driver's gaze direction and head pose, the eyelid opening signal, and corresponding quality signals. Additional information cannot be published due to a non-disclosure agreement.



(a) Driver seat of the testing vehicle

(b) Fovio camera system

Figure 2.6: On the left, an overview of the inside of the testing vehicle is shown. On the right, the mounting position of the camera system is shown.

A *MicroAutoBox*[9] (MABX) is connected to the CAN bus of the vehicle and enables the testing vehicle to execute compiled Simulink models in real-time on an IBM PPC 750GL, 900MHz processor. Further, a FleetPC-7 Car-PC with an Intel Core i7 2710QE was con-

[8] Complementary metal-oxide-semiconductor

[9] Realtime hardware system for functional prototyping produced by dSpace GmbH.

nected with the MABX enabling the development of Simulink models in MATLAB 2012 directly in the vehicle and guaranteed synchronicity of the Dikablis and *laserBird* signals transmitted to the CAN bus. The variables of the running Simulink model on the MABX can be visualized on graphical user interfaces (GUI) of the software *ControlDesk 5.1*[10]. All CAN bus signals were recorded by means of ControlDesk 5.1 with a sampling rate of 50 Hz.

2.4.3 Pre-Study NEBAF

As a pre-study, 85 experiments were conducted in a detailed Mercedes-Benz E-class (W212) mounted in the moving base simulator described in Section 2.4.1. In line with internal terminology, the study will be referred to as *NEBAF*[11]. For this study, a 35 minute-long route on a German highway with two to three lanes was simulated. At the beginning of each experiment, the subjects drove manually followed by a conditionally automated driving section without secondary tasks to introduce the subjects to the simulator. Both introductory route sections were about one minute long. After this introductory drive, the subjects were asked to perform secondary tasks on a touch screen mounted in front of the center console in the cabin. In total, there were four different secondary tasks to perform by means of the touch screen: reading news, watching a short video, writing an e-mail, and listening to music. After finishing a given task, the subjects had to independently select the next task manually over the GUI provided on the touch screen. Furthermore, eleven of the total 85 subjects were part of a control group, which means that they did not perform any secondary tasks on the touch screen. The eye movements of the subjects were recorded by means of the Dikablis Essential Glasses 2.3.4 while the head movements were measured by means of the *laserBird* laser scanner 2.3.2. The recorded data was used for investigating the eye movement behavior during conditionally automated driving scenarios in Section 3, where details on the distribution of the subjects are mentioned.

2.4.4 Experiment Ko-HAF

The main driving simulator study was conducted in the Mercedes-Benz moving base driving simulator described in Section 2.4.1. Since this study was combined with a driving simulator study of the Ko-HAF[12] project, the study will be referred to as KoHAF. In total, 112 subjects drove in a detailed Mercedes-Benz E-class (W212) equipped with a conditionally automated driving function. The interior of the vehicle was equipped with three cameras monitoring the footwell, the driver's face, and the steering wheel and center console region as shown in Figure 2.7. To display the current state of the automated driving function, a 6″ monitor was mounted to the left of the steering wheel. The four possible states of the automated driving function include: *Automated function available*, *Automated function not available*, *Automated function active*, and *Take-Over*. While activated, the automated driving function takes over the lateral and longitudinal control as

[10] Software of the dSpace GmbH for the development and evaluation of control devices.

[11] NEBAF=Nebentätigkeiten beim hochautomatisierten Fahren

[12] Ko-HAF=Kooperatives hochautomatisertes Fahren

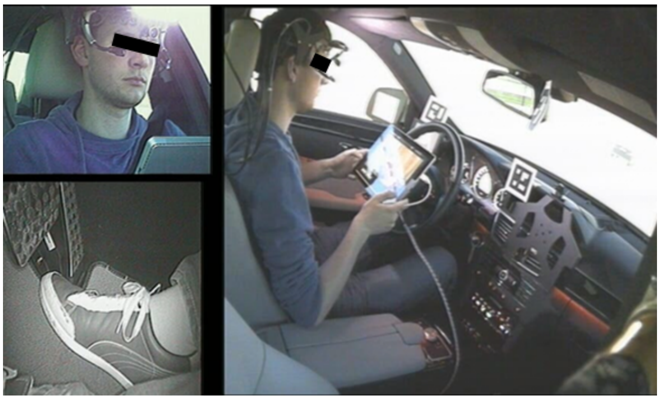


Figure 2.7: Test subject watching a video on the tablet while being monitored by the three installed cameras.

well as the responsibility for the vehicle. The usage and behavior of the automated driving function were explained thoroughly to each subject before the experiment. The simulated route represented a German highway with three lanes and no speed limits. However, the automated driving function was implemented so that the constant speed was 130 Km/h. To acclimate the drivers, they were told to first drive manually for about four minutes, then follow a conditionally automated route section without any secondary tasks for about three minutes. After this introduction, the tablet was activated and the subject was asked to perform secondary tasks.

The tablet was either mounted at the center console to represent an integrated system or was freely moveable. Each subject was shown how to use the tablet before the actual experiment and was thus able to independently access each subsequent task. Possible secondary tasks were *reading news* or *watching a video*. Moreover, there were route sections where the subject did not perform any tasks at all. The secondary task was defined as *idle* for these sections. In addition, there was a control group performing no secondary tasks for the complete experiment. During the experiment, the subjects experienced three different take-over situations due to missing lane marks with a take-over time of 3.5 seconds. The take-over was indicated by an acoustic and visual warning. The first take-over occurred on a straight after five minutes of automated driving and did not require any specific actions from the driver. This simple take-over situation was also used to let the driver experience his or her first take-over situation. This situation will be referred to by the term *Straight*. The second and third take-over situations were both generated after about eight minutes of automated driving and were permuted over the different subjects. One of these situations took place in a left curve while at the same moment a 7 m/s cross-wind from the left was simulated for seven seconds. This situation required the driver to hold the vehicle inside the lane by a lateral intervention. The other take-over was simulated on a straight with high traffic density on the adjacent left lane. Moreover, one vehicle cut in 20 m in front of the subject vehicle in the far right lane and decelerated from 126 Km/h to about 80 Km/h. In this situation, the driver had to recognize the braking maneuver of the leading vehicle as well as the blocked adjacent lane and had to brake appropriately. These situations will be referred to by the terms *Braking* and *Cross-wind*. Note that no artificial distraction was performed before the take-over situations to provide a natural behavior of the drivers, e.g., allowing interruptions of the secondary task at any given moment.

Eye movements were recorded by means of the Dikablis professional eye tracker as described in Section 2.3.4 whereas the in Section 2.3.2 described *laserBird* measured head position and rotation. Both sensors sent their signals to an installed PC, which transmitted these signals on the CAN bus of the vehicle and guaranteed sensor synchronicity. All CAN bus signals were recorded with a sampling rate of 100 Hz. The gaze direction was estimated by incorporating the head pose of the laser scanner and the head-mounted eye-tracking. To ensure a smooth experimental process, both the eye tracker and the head-mounted sensor of the laser scanner were attached to a helmet. To incorporate the head- and eye-tracking to generate a gaze direction, a calibration process was conducted at the beginning of each experiment. This calibration process and the procedure to fuse the head- and eye-tracking signals to a gaze direction will be described more detailed in Section 2.5.

2.4.5 Conditionally Automated Real Driving Study

To gather real-world driving data of realistic driver behavior in conditionally automated scenarios with a close-to-production driver camera, a driving study by means of the testing vehicle described in Section 2.4.2 was conducted. The study took place on the test track of the Daimler AG in Sindelfingen, Germany shown in Figure 2.8. The test track contains two straights of about 1000 m connected by two steep turns. Distronic Plus with Steering Assist was tested countless times on the straights before the experiment to ensure the correct and faultless behavior of the system. As a result, it was possible to simulate conditionally automated driving on both straights. However, the two steep turns prevented the use of the Distronic Plus with Steering Assist, thus manual driving becomes necessary. In the following, the acronym *DTR+Q* will refer to the Distronic Plus with Steering Assist. According to the procedure of the study, two subjects were necessary for each experiment.

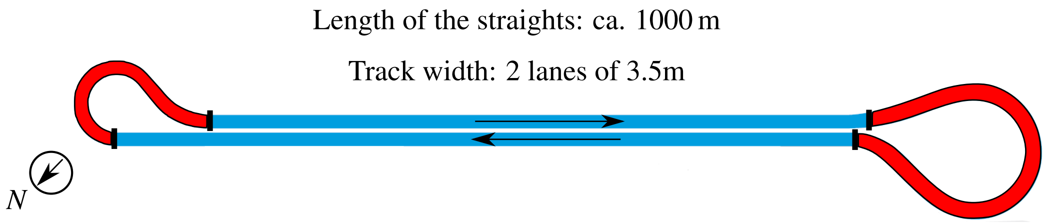

Length of the straights: ca. 1000 m
Track width: 2 lanes of 3.5m
N

Figure 2.8: Layout of the test track of the Daimler AG in Sindelfingen, Germany. Route sections highlighted in blue allowed simulating conditionally automated driving. Route sections highlighted in red require manual driving.

One subject was driving a leading series vehicle supported by an ACC. The subject was told to set the ACC to a velocity of 30 Km/h and 50 Km/h on the eastbound and westbound straight, respectively. In addition, the driver varied the velocity on the eastbound straight four-times at self-chosen route sections. In detail, the driver was told to set the ACC velocity to 40 Km/h and reduce it to 30Km/h again as soon as a velocity of 39 Km/h on the digital speedometer was displayed at these four route sections. This driving profile was used to simulate stop-and-go traffic and provoke gazes at the road. For demonstration purposes, this procedure was performed in an introductory lap with both drivers in the vehicle. Note that both subjects were instructed at the beginning of the experiment with this procedure to prevent different levels of preknowledge. The subject of the leading vehicle drove according to these instructions until the instructor stopped the experiment via walkie-talkie.

The second subject drove the testing vehicle and followed the leading vehicle at all times. At the beginning of both straights, DTR+Q was activated to simulate a conditionally automated drive. In addition, the driver had to perform specific tasks such as reading a text, watching a video, or being idle. Tasks were performed on the tablet mounted at the center console or on a handheld device with a 7″ display. The exact procedure was as follows:

Lap	Route section	Task
1	eastbound	activate DTR+Q reading a text on the hands-free device
	westbound	follow manually without DTR+Q
2	eastbound	activate DTR+Q watching a video on the hands-free device
	westbound	activate DTR+Q no specific task to perform (idle)
3	eastbound	activate DTR+Q reading a text on the handheld device
	westbound	activate DTR+Q looking successively into the areas... • windshield/on the road • left exterior mirror • right exterior mirror • interior mirror • mounted tablet
4	eastbound	activate DTR+Q watching a video on the handheld device
	westbound	activate DTR+Q no specific task to perform (idle)

Table 2.2: The procedure for the subject driving the testing vehicle is summarized.

The duration of the eastbound and westbound route section was about 102 s and 70 s. The listed instructions were given to the driver by the instructor while a rater labelled online the performed secondary tasks and the driver's gazes at the road. Instructor and rater were riding along on the back seat of the testing vehicle. Further, the instructor asked the subject to resume the control of the vehicle at the end of the straight during the curves. The tasks were permuted over the different subjects, i.e. depending on the subject the tasks *reading* and *watching a video* were first performed on the handheld or hands-free device, respectively. After one iteration of the experiment, the subjects changed the vehicles and the experiment was conducted a second time. Due to safety regulations of the test vehicle and track, only experienced drivers participated in the study who are familiar with the test track and its rules. In total, ten male subjects[13] with normal or corrected visual acuity participated in the driving study. However, only nine of the originally ten drivers can be used for the later evaluations based on these data sets since the near-to-production camera system was only sporadically able to detect the eyes of one driver and, therefore, only an extremely fragmentary signal of the gaze direction is available.

2.5 Fusion of Head- and Eye-Tracking Devices

Some of the later approaches require the driver's gaze direction as input to determine the secondary task or the gazes at the road. For simplicity, this study assumes a *mid-eye model* which is defined by a head model with one eye located between the two real eyes at the nose bridge of the subject. Hence, an averaged gaze direction originating at the mid-eye is used for calculations instead of two separated gaze directions for each eye. In case of the real driving study described in Subsection 2.4.5, the mounted Fovio camera already provides a gaze direction represented by the normalized 3-dimensional vector

$$\mathbf{\Phi}_{gaze,cart} = \begin{bmatrix} x_{gaze} \\ y_{gaze} \\ z_{gaze} \end{bmatrix} \quad \text{with } x_{gaze}, y_{gaze}, z_{gaze} \in [-1,1] \subset \mathbb{R} \tag{2.12}$$

in the Cartesian coordinate system originating at the position of the camera as shown in Figure 2.9(a). The transformation of the Cartesian gaze vector into a gaze vector of a spherical coordinate system is given by the equations

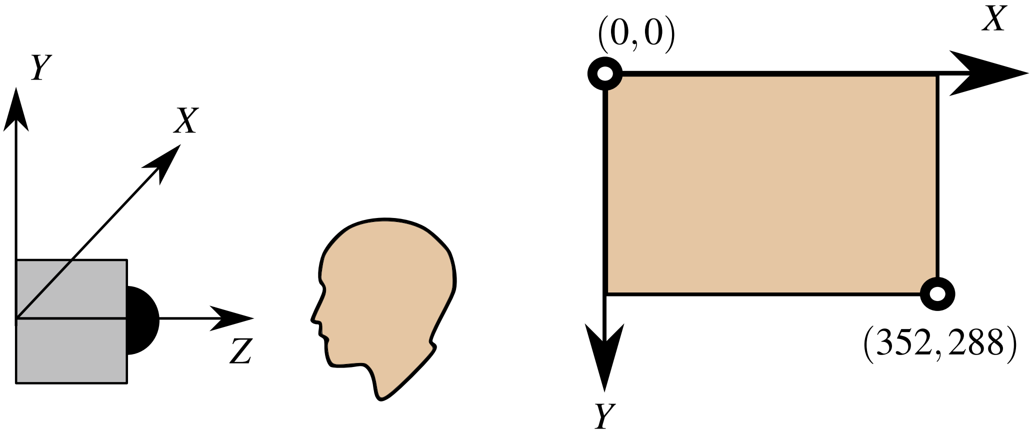
$$\theta = arccos\left(\frac{z_{gaze}}{\sqrt{x_{gaze}^2 + y_{gaze}^2 + z_{gaze}^2}}\right) \quad [rad] \tag{2.13}$$

$$\psi = arctan\left(\frac{y_{gaze}}{x_{gaze}}\right) \quad [rad] \tag{2.14}$$

$$\mathbf{\Phi} = \begin{bmatrix} \psi \\ \theta \end{bmatrix} \tag{2.15}$$

where the angle ψ will be the yaw angle and the angle θ will be the pitch angle of the subject's gaze. The *laserBird* head tracker and Dikablis eye tracker which were applied in the conducted KoHAF experiment do not provide any gaze direction but usually highly

[13] mean age of 38 years (range 27-52, SD=9)



(a) Cartesian world coordinate system of the Fovio camera. The positive direction of the horizontal X axis is to the right from a driver's view into the camera, the positive direction of the vertical Y axis is up from a driver's view into the camera, and the positive direction of the Z axis is towards the driver.

(b) 2-dimensional pixel coordinate system of the Dikablis eye camera with a resolution of (352×288). Note that the positive direction of the Y axis is down from a driver's view into the camera.

Figure 2.9: Visualization of the Fovio camera's coordinate system and of the Dikablis eye camera's pixel coordinate system.

accurate signals. The Dikablis camera system generates 2-dimensional pixel coordinates

$$\mathbf{\Phi}_{DK} = \begin{pmatrix} eye_x \\ eye_y \end{pmatrix}, \; eye_x \in [0,352] \subset \mathbb{N}; \; eye_y \in [0,288] \subset \mathbb{N} \tag{2.16}$$

describing the location of the center of the driver's pupil mapped on the eye camera's coordinate system shown in Figure 2.9(b). To transform these pixel coordinates into angles representing the rotation of the driver's eyes, a function

$$f : \begin{pmatrix} eye_x \\ eye_y \end{pmatrix} \to \begin{pmatrix} \psi_{eye} \\ \theta_{eye} \end{pmatrix} = \mathbf{\Phi}_{eye} \tag{2.17}$$

is required. In this section, all necessary steps to estimate such a function f and the gaze direction will be discussed. First, the calibration routine at the beginning of the experiments is discussed in Subsection 2.5.1 which generates the necessary data samples for estimating the function f. However, the eye tracker suffers from shortcomings of the provided pupil detection algorithm, occurring signal noise, and torsions of the eye camera. Hence, these disturbances will be corrected in Subsection 2.5.2 and 2.5.3 before the actual head pose and finally the driver's gaze can be estimated in Subsection 2.5.4 and 2.5.5.

2.5.1 Calibration Process

A calibration routine was performed at the beginning of each experiment of the KoHAF study. For this calibration, the subject was seated on the driver's seat of the driving simulator's vehicle and was asked to check and if necessary adjust the seat and each mirror according to individual body size. Afterwards, the subject was asked to put on the helmet with the attached head sensor of the *laserBird* and the Dikablis Professional glasses. For the sake of efficiency, the different steps of the calibration process had been explained to each subject and they were enabled to put on the helmet for testing purpose in the preliminary discussion of this experiment. Simultaneously, this procedure was monitored by the examiner who afterwards started the recording of all Dikablis, *laserBird*, and vehicle-based signals. The calibration serves to determine individual differences between different subjects, e.g., variations of the location of the eye camera or the *laserBird*'s head sensor, and to collect data samples for the later mapping process. For the calibration, the subject had to focus a given target inside or outside the vehicle. At the same time, the subject rotated their head randomly, e.g., they shook their head, nodded, or combined these movements at various angles for about 12 s. The general idea shown in Figure 2.10 is that the eyes will rotate in the opposite direction of the head rotation assuming the subject keeps focusing on the target. Moreover, the pupil's rotation equals the angle between mid-eye and reference point after subtracting the head pose. Hence, this calibration is based on the vestibulo-ocular reflex described in Subsection 2.2.2. This procedure was repeated for each target. In total, there were three different targets

$$\mathcal{V}_{\text{ref1}}, \mathcal{V}_{\text{ref2}}, \mathcal{V}_{\text{ref3}} \in \mathbb{R}^{3\times 1} \quad [\text{mm}] \tag{2.18}$$

as shown in Figure 2.11. The target $\mathcal{V}_{\text{ref1}}$ was projected on the screen of the driving simulator directly in front of the driver at a distance of about 3.5 m. The second target $\mathcal{V}_{\text{ref2}}$ was visualized in the center of the left mirror while the last target $\mathcal{V}_{\text{ref3}}$ was at the upper left corner of the center display. The locations of the targets were known precisely as 3-dimensional computer-aided design (CAD) coordinates or were manually measured with respect to the vehicle coordinate system. During the performance of the head movements, the driver pressed a button on the steering wheel so that the calibration intervals could later be assigned to the correct signal parts. Note that there was no additional calibration of the exact location of the *laserBird*'s head sensor, since that is an extremely time-consuming and complicated step. Hence, the vector

$$\mathbf{lb}_{nose} \in \mathbb{R}^{3\times 1} \quad [\text{mm}] \tag{2.19}$$

describing the offset from the head sensor to the mid-eye was estimated based on the data of some measurements performed before the KoHAF study.

2.5.2 Signal Pre-Processing

In a first step, the signals provided by the Dikablis software have to be rearranged since a timestamp provided by Dikablis reveals that the samples were not transmitted to the

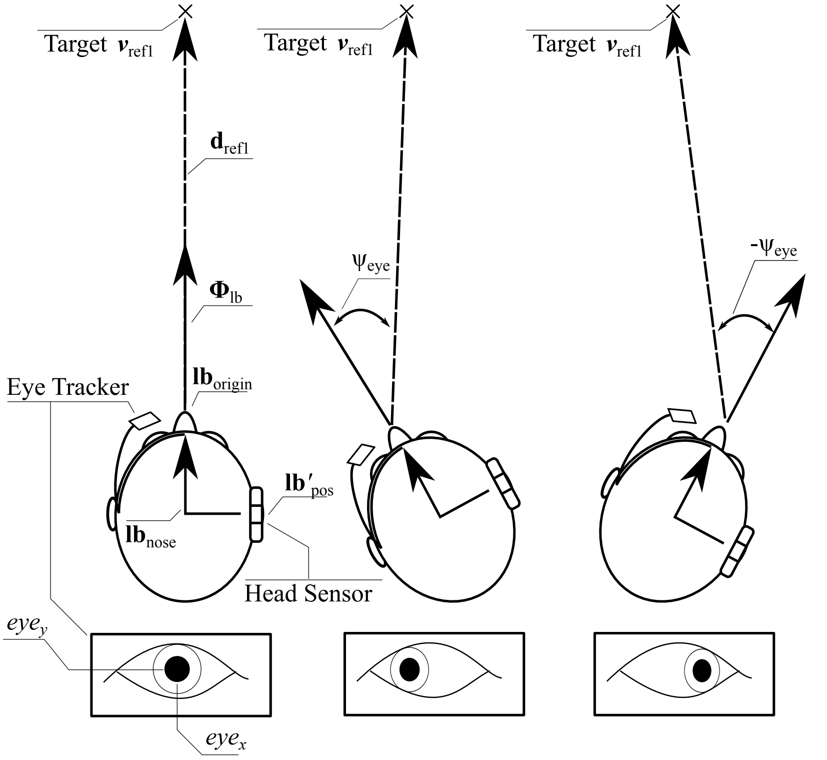

Target $\boldsymbol{v}_{\text{ref1}}$
Target $\boldsymbol{v}_{\text{ref1}}$
Target $\boldsymbol{v}_{\text{ref1}}$
$\mathbf{d}_{\text{ref1}}$
$\boldsymbol{\Phi}_{\text{lb}}$
ψ_{eye}
$-\psi_{\text{eye}}$
Eye Tracker
$\mathbf{lb}_{\text{origin}}$
$\mathbf{lb}'_{\text{pos}}$
$\mathbf{lb}_{\text{nose}}$
Head Sensor
eye_y
eye_x

Figure 2.10: Calibration procedure by means of random head movements. As long as the driver's eyes stay focused on the corresponding target, the eyes rotate the same distance as the head but in the opposite direction. Here, the subject shook their head to the left and right.

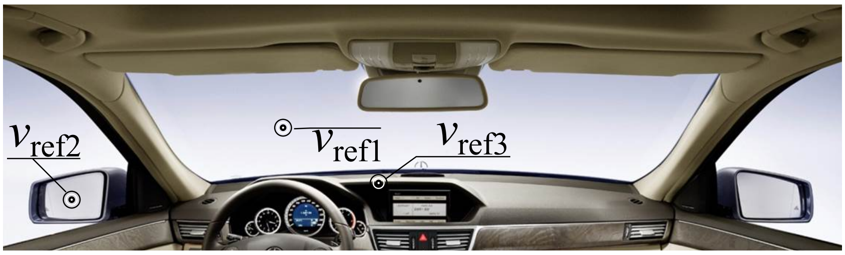


Figure 2.11: Locations of the targets of the three calibration steps.

CAN bus in their order of recording. This error is most likely caused by non-synchronized threads[14] of the pupil detection algorithm which experience individual delays. However, the correct order can be restored by means of the timestamp.

The applied pupil detection algorithm is a major factor influencing the amount of false measurements. Depending on the orientation of the camera, the lighting conditions, the

[14] Simultaneous program sections

individual eye structure, parts of the eye or face structure could be falsely detected as pupil. This error generates high-frequency noise in the position signals eye_x and eye_y, but also in other signals describing the structure of the detected pupil, e.g., the width or height of the pupil. Moreover, the Dikablis software provides a validation signal highlighting samples where the pupil detection was successful. Note that a successful pupil detection only indicates that an area was found that is similar to the sought pupil. However, no warranty is given that this area is truly the eye's pupil. To handle these false detections, a filter is applied to delete all samples outside of the band defined by the twofold standard deviation in relation to the median of the horizontal and vertical pupil position as well as the pupil height and width. Moreover, the signals are made plausible by deleting signal steps which indicate an eye velocity of more than a pre-defined threshold. The threshold is an experience value of $2000\,px/s$ which was chosen explicitly for the data set of the KoHAF study. Afterwards an erosion with a temporal window of 40 ms is performed to delete short signal parts in which less than five sequential, valid pupil detections had been performed. Finally, the signal gaps resulting from the previously described filtering steps will be refilled via a *Nearest Neighbor Interpolation*[15]. Thus, correctly detected saccades will be restored while the deleted outlier will be interpolated. As a last step, the interpolated signal is smoothed by an averaging filter with a window size of 100 ms.

2.5.3 Removal of Torsions of the Eye Camera

The Dikablis eye camera has to be adjusted individually to each subject's head and face structure. This can lead to unwanted torsions of the camera about the optical axis resulting in the rotation of the horizontal and vertical axis as shown in the upper part of Figure 2.12. To estimate and remove such rotations, the maximum variance of the recorded 2-dimensional eye-tracking data is searched for. This approach is based on the assumption that the variance of the horizontal axis is the greatest in the eye-tracking data. If the camera was rotated during recording, the x and y axis are oblique-angled. To determine a possible rotation of the camera an orthogonal transformation is sought which maximizes the variance of the two factors eye_x and eye_y. For this purpose, an approach for *Factor Analysis*[16] called *Varimax* is applied which rotates the orthogonal basis of the given coordinate system until the maximum variance of the squared loadings is reached. Mathematically formulated, the function

$$\sum(q_{j,l}^2 - \bar{q}_{j,l}^2)^2 \tag{2.20}$$

needs to be maximized where $q_{j,l}^2$ represents the squared loading of the j-th variable on the l-th factor and $\bar{q}_{j,l}^2$ represents the mean of the squared loadings. To solve this equation, a *Principal Component Analysis*[17] can be applied. As shown in Figure 2.12, the torsion of

[15] Approximation of a missing point depending on the closest sample points around this value.

[16] Collection of statistical methods to infer from observed variables to unobserved factors.

[17] Statistical method to transform possible correlated variables on less, more significant estimated linear combinations which are called principal components.

the eye camera can be eliminated and the data is transformed onto the original coordinate system.

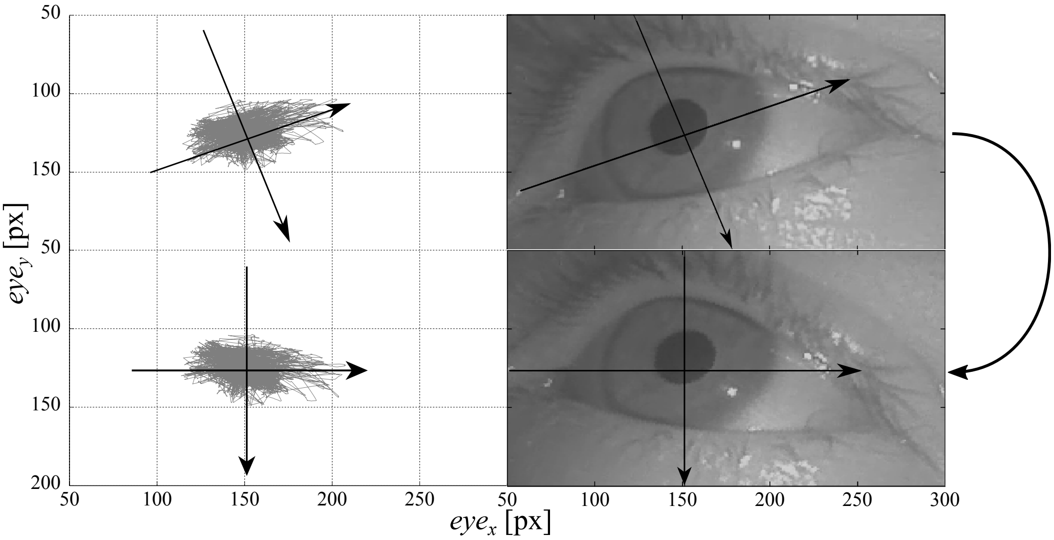


Figure 2.12: Example of a recording including camera torsion. The torsion can be eliminated using *Varimax*.

2.5.4 Estimation of the Driver's Head Pose

All output signals provided by the *laserBird* refer to its own coordinate system originating at $\mathbf{lb}_{init}$ with a positive x axis pointing to the driver's seat, a positive y axis pointing to the front of the vehicle, and a positive z axis pointing downwards. In contrast to the *laserBird* coordinate system, the vehicle coordinate system has its origin at the center of the front axle where the positive direction of the x axis points to the back of the vehicle, the positive direction of the y axis points to the right side of the vehicle, and the positive direction of the z axis points upwards. However, the *laserBird* coordinate system differs from the vehicle coordinate system not only in terms of the origin and direction of the axis but also with regard to the terminology of the corresponding rotations. Both coordinate systems and the corresponding terminology are shown in Figure 2.13.

To avoid switching between these coordinate systems, the calculated head position and rotation shall be transferred to the vehicle coordinate system. First, the axis of the *laserBird* coordinate system will be adapted to the axis of the vehicle and the mounting position of the laser scanner will be added to the origin of the vehicle coordinate system by

$$\mathbf{lb}'_{pos} = \begin{pmatrix} -lb_{pos_y} \\ -lb_{pos_x} \\ -lb_{pos_z} \end{pmatrix} + \mathbf{lb}_{init} \quad [\text{mm}]. \tag{2.21}$$

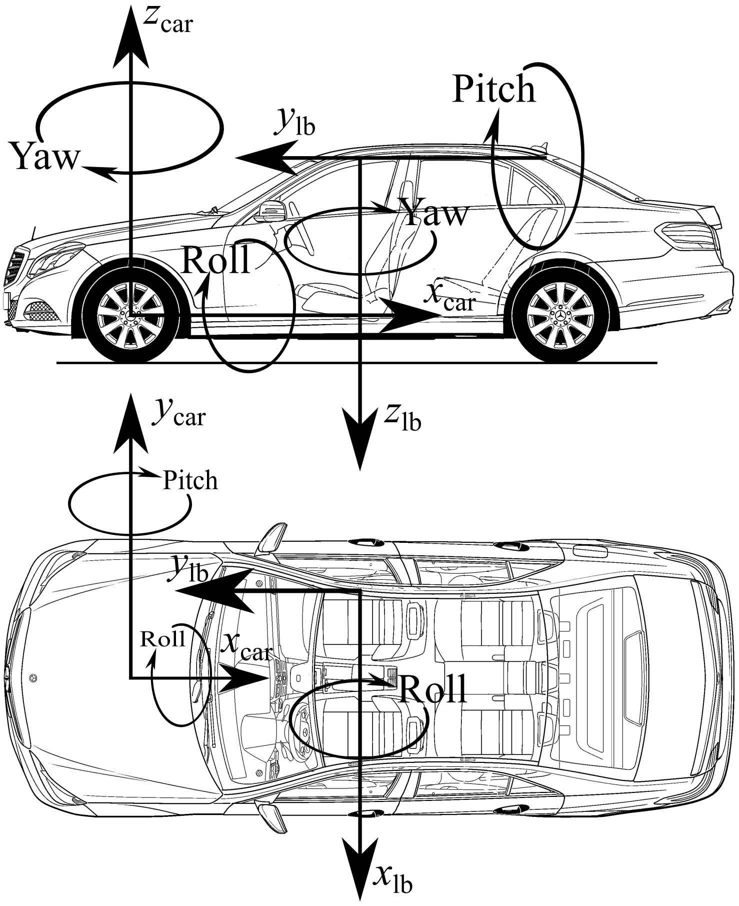


Figure 2.13: Vehicle and *laserBird* coordinate system including the rotation directions.

Originally, the *laserBird* rotated its signals about the x axis, then about the y axis, and finally about the z axis. For simplicity, this rotation order will be referred to as XYZ. Since the *laserBird* coordinate system was adapted to the vehicle coordinate system by interchanging and inverting the axis, the rotation order changed to YXZ with reverse rotation direction compared to the vehicle coordinate system. In the following, the rotation vector $\mathbf{lb}_{rot}$ shall be adapted to the vehicle coordinate system and its corresponding rotation order. Therefore, the three Euler[18] angles[19] of $\mathbf{lb}_{rot}$ defined in (2.11) shall be named according to

- ϕ_{lb}: rotation angle about the x axis (roll)
- θ_{lb}: rotation angle about the y axis (pitch)
- ψ_{lb}: rotation angle about the z axis (yaw)

[18] Leonhard Euler, * 15. April 1707 in Basel, Switzerland, † 18. September 1783 in St. Petersburg, mathematician and physicist who belongs to the pioneers in the field of analysis and number theory and provided significant contributions in mechanics, algebra, and graph theory.

[19] Three parameters to describe the orientation in a rigid, three-dimensional space.

and the basic rotation matrices shall be defined as

$$\mathbf{R}_x(\phi_{lb}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_{lb}) & -\sin(\phi_{lb}) \\ 0 & \sin(\phi_{lb}) & \cos(\phi_{lb}) \end{pmatrix} \tag{2.22}$$

$$\mathbf{R}_y(\theta_{lb}) = \begin{pmatrix} \cos(\theta_{lb}) & 0 & \sin(\theta_{lb}) \\ 0 & 1 & 0 \\ -\sin(\theta_{lb}) & 0 & \cos(\theta_{lb}) \end{pmatrix} \tag{2.23}$$

$$\mathbf{R}_z(\psi_{lb}) = \begin{pmatrix} \cos(\psi_{lb}) & -\sin(\psi_{lb}) & 0 \\ \sin(\psi_{lb}) & \cos(\psi_{lb}) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.24}$$

The combined rotation matrix $\mathbf{R}_{zyx}$ with the order ZYX is given by

$$\mathbf{R}_{zyx}(\phi_{lb}, \theta_{lb}, \psi_{lb}) = \mathbf{R}_x(\phi_{lb}) \cdot \mathbf{R}_y(\theta_{lb}) \cdot \mathbf{R}_z(\psi_{lb}). \tag{2.25}$$

Note that these 3-dimensional rotations are not commutative, i.e. the order of the matrix multiplication is essential. In a first step, the rotation angles have to be interchanged according to the permutation of the axis of the coordinate system. This can be accomplished through the matrix multiplication

$$\mathbf{lb}'_{rot} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \cdot \mathbf{lb}_{rot} \tag{2.26}$$

which interchanges the roll and pitch angle and inverts all three angles. The three Euler angles given in the vehicle coordinate system can then be determined via the equations

$$\phi_{veh} = \text{atan2}(\sin(\phi_{lb}), \cos(\phi_{lb}) \cdot \cos(\theta_{lb})) \tag{2.27}$$

$$\theta_{veh} = -\arcsin(-(cos(\phi_{lb}) \cdot \sin(\theta_{lb}))) \tag{2.28}$$

$$\begin{aligned} \psi_{veh} = \text{atan2}(&cos(\theta_{lb}) \cdot \sin(\psi_{lb}) + \sin(\phi_{lb}) \cdot \sin(\theta_{lb}) \cdot \cos(\psi_{lb}), \\ &\cos(\theta_{lb}) \cdot \cos(\psi_{lb}) - \sin(\phi_{lb}) \cdot \sin(\theta_{lb}) \cdot \sin(\psi_{lb})) \end{aligned} \tag{2.29}$$

where atan2 is the arctangent with two input arguments.

Since it is assumed that the head pose has its origin at the nose bridge between the subject's eyes, the distance $\mathbf{lb}_{nose}$ between the *laserBird* head sensor and the nose bridge needs to be added to the head position $\mathbf{lb}'_{pos}$

$$\mathbf{lb}_{origin} = \mathbf{lb}_{nose} + \mathbf{lb}'_{pos} \quad [mm]. \tag{2.30}$$

As described in Subsection 2.5.1, the vector $\mathbf{lb}_{nose}$ was estimated based on data from a pre-study. Measuring $\mathbf{lb}_{nose}$ exactly would have been too time-consuming during the experiment. However, since the head sensor was attached to a helmet the individual differences of the mounting position of the sensor can be seen as restricted. Note that $\mathbf{lb}_{nose}$ has to be rotated by $\mathbf{R}_{zyx}(\phi_{lb}, \theta_{lb}, \psi_{lb})$ to match the head rotation in the vehicle coordinate system. The direction of the head pose can be defined by rotating a normalized vector along the neutral x axis by means of the combined rotation matrix

$$\mathbf{lb}_{dir} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \cdot \mathbf{R}_{zyx}(\phi_{lb}, \theta_{lb}, \psi_{lb}) \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{2.31}$$

The *laserBird* head sensor was rotated along the y axis for mounting it on the helmet. Thus, the head direction is rotated in the original state by the left matrix multiplication of equation (2.31). The final head pose can be visualized as a vector with the origin at $\mathbf{lb}_{origin}$ and the direction $\mathbf{lb}_{dir}$

$$\mathbf{\Phi}_{lb} = \mathbf{lb}_{origin} + k \cdot \mathbf{lb}_{dir} \tag{2.32}$$

with $k > 0$.

2.5.5 Estimation of the Driver's Gaze Direction

If it is assumed that the eyeball approximates a sphere with a punctate location of the pupil center on its surface, $\mathbf{\Phi}$ represents the amount of angles which can be reached by the movement of the eye. These angles are mapped on the 2-dimensional coordinate system of the Dikablis eye camera and need to be transformed back over the function $f(\mathbf{\Phi}_{eye})$ to the spherical angles of the eyeball. For this purpose, the described calibration step was performed to gather eye- and head-tracking data of many different angles of the pupil center. The eye-tracking data is given as the coordinates eye_x and eye_y. Further, the rotation of the pupil can be deduced by

$$\mathbf{d}_{refx} = (\mathcal{V}_{refx} - \mathbf{lb}_{origin}),\ x \in \{1,2,3\} \tag{2.33}$$

$$\mathbf{\Phi}_{eye,cart} = \mathbf{d}_{refx} - \Phi_{lb} \tag{2.34}$$

and then transformed to spherical yaw ψ_{eye} and pitch angle θ_{eye} by equation (2.14) and (2.13). These pairs of camera coordinates and eye pupil rotations $(\mathbf{\Phi}_{DK}, \mathbf{\Phi}_{eye})$ are applied to estimate f. The function is approximated by half of a sinusoidal curve generated by the inverse trigonometric function arcsine over

$$\begin{pmatrix} \arcsin\left(\frac{\hat{eye}_x}{\sqrt{1-\hat{eye}_y^2}}\right) \\ \arcsin(\hat{eye}_y) \end{pmatrix} \tag{2.35}$$

with $\hat{eye}_x$ and $\hat{eye}_y$ as camera coordinates scaled to the interval $[-1,1]$. In Figure 2.14, it can be seen that by applying the arcsine it is possible to reflect the characteristic of eye cameras to resolve smaller angles on a wider range of pixel coordinates whereas the resolution of large pupil angles decreases. Figure 2.15 visualizes an exemplary 3-dimensional plain representing function $f(\mathbf{\Phi}_{DK})$ for one test subject. The 99%- and 1%-quantiles of $\mathbf{\Phi}_{eye}$ and $\mathbf{\Phi}_{DK}$ were chosen as grid points for the arcsine which increases the robustness of the approach against outliers, e.g., due to temporal loss of focus. However, as it can be seen in Figure 2.15, the resolution of the pitch angle only covers a range from $[-20°, 20°]$. The reason for that is the individual rotation range during the calibration process. As described in Subsection 2.5.1, the subjects were asked to perform random head rotations of different pitch and yaw angle size. However, most subjects focused on varying the size of the yaw angle while the pitch angle only covered a small angle range. In addition, there were some subjects who performed only small head movements for both angles. Such performances result in less or even no data for large head and pupil rotations. Consequently, function f cannot be estimated over the same angle range for all subjects and has to be extrapolated

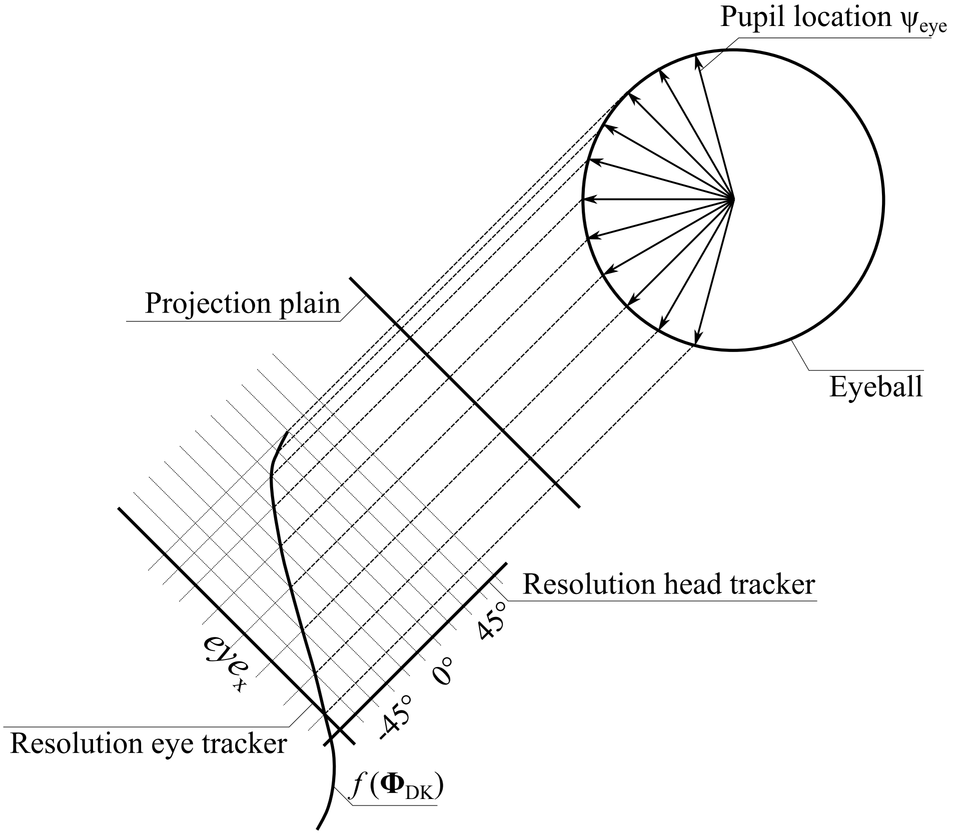


Figure 2.14: Functional illustration of the mapping process for different horizontal pupil angles on the corresponding eye camera coordinates. In this topview, the sinusoidal curvature of function f can be seen.

in these cases.

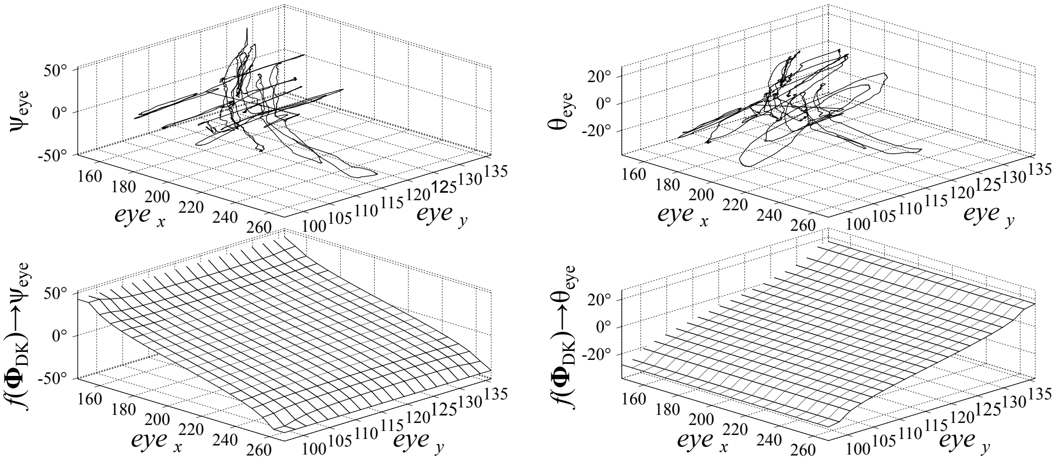


Figure 2.15: Example of an estimated function $f(\mathbf{\Phi}_{DK})$ plotted as 3-dimensional plain. The upper plots visualize the raw data recorded during the calibration phase. The lower plots visualize the final transformation function after applying all the mentioned steps in Section 2.5.

After estimating the function f for calculating the angles of the pupil based on the recorded coordinates of the eye camera, the final gaze $\mathbf{\Phi}_{cart}$ can be calculated by

$$\mathbf{\Phi}_{gaze,cart} = \frac{\mathbf{lb}_{dir} + \mathbf{dk}_{dir}}{||\mathbf{lb}_{dir} + \mathbf{dk}_{dir}||} \tag{2.36}$$

where $\mathbf{dk}_{dir}$ represents the direction of the eye gaze without head pose calculated over the equation

$$\mathbf{dk}_{dir} = \begin{pmatrix} \sin(\theta_{eye}) \cdot \cos(\psi_{eye}) \\ \sin(\psi_{eye}) \cdot \sin(\theta_{eye}) \\ \cos(\theta_{eye}) \end{pmatrix}. \tag{2.37}$$

The corresponding gaze angles as spherical coordinates can be calculated by equation (2.13) and (2.14).

3 Eye Movement Classification in the Context of Conditionally Automated Driving

In the context of automated driving, especially while the driver is performing various secondary tasks, the eye movement behavior is assumed to vary significantly due to task- and inter-individual differences. However, up until now there is no verification of this assumption. Such variations would entail particular challenges for the automated eye movement classification methods and, therefore, imply the development of adapted approaches for the automated driving context. In this chapter, a method is presented that explicitely addresses variations in the eye movement behavior due to task- and inter-individual differences such as during conditionally automated driving scenarios.

In a first step, state-of-the-art automated eye movement classification is summarized and discussed with regard to its adaptability to changes in the eye movement behavior in Section 3.1. The work indicates that current classification methods are not able to adapt sufficiently to changes in eye movement behavior in automated driving scenarios. Thus, a novel algorithm named MERCY is introduced and evaluated in Section 3.2. Besides the application as an eye movement classification algorithm, MERCY can be applied to analyze the variations in eye movement behavior in general. A thorough examination of eye movement behavior for conditionally automated driving scenarios is performed in Section 3.3 based on the data of the pre-study NEBAF described in Section 2.4.3. The chapter concludes with the description and application of MERCY in the testing vehicle introduced in Section 2.4.2. The results from the author's publication [12] provide the essential content of this chapter.

3.1 Methods for Eye Movement Classification

Various methods for classifying the eye movements relevant for this work, namely saccades and fixations, can be found in the literature. However, only a few of these algorithms were especially designed with a driving context and none of them were applied and reported in a conditionally automated setting. Furthermore, there are multiple approaches for categorizing these methods with regard to different aspects such as the type of applied thresholds. The following subsections provide an overview of these methods and introduce the existing taxonomies, leading to the unresolved question of the adequate choice of the classification threshold.

3.1.1 Previous Work

An overview of different approaches for eye movement classification is shown in Table 3.1, including, among others, two taxonomies introduced by Salvucci and Goldberg in [55] and by Kasneci et al. in [56]. Salvucci and Goldberg introduced a novel taxonomy to realize a first categorization of automated classification methods consisting of a minimal set of three spatial criteria, namely *velocity-*, *dispersion-*, and *area-based* criteria, and two temporal criteria, namely *duration sensitive* and *locally adaptive* criteria [55]. The *locally adaptive* criterion refers to the analysis of temporally adjacent data points, while *duration sensitivity* implies the incorporation of the duration of the respective eye movement. Further, the particular advantages and disadvantages of the chosen categories were evaluated in [55] by means of five representative algorithms. The categories were named after the spatial criteria and included *velocity-based*, *dispersion-based*, and *area-based* methods. *Dispersion-based* methods are based on the fact that data points of fixations tend to build clusters, while larger outliers usually belong to saccades. Such methods tend to provide adequate results, but need to be adjusted carefully to the respective task. A typical algorithm of this category is *I-DT*[1] originally introduced by Widdel in [57] and analyzed in [55], which incorporates the dispersion threshold with a duration threshold. A more sophisticated method is given by *I-MST*[2] introduced by Goldberg and Schryver [58], which detects saccades based on the branching depths of a constructed MST. *Velocity-based* algorithms separate saccades and fixations by analyzing the velocity of the sequential data points. High velocities imply the affiliation of these data points to saccades while low velocities indicate fixations. Approaches based on the velocity profile are fast, online-and real-time capable, and easy to implement, but often struggle with noise due to their fixed threshold. *I-VT*[3] is named as a representative algorithm in [55] requiring the specification of only one fixed parameter defining the velocity threshold. An adaptive option for a velocity-based algorithm is given by *I-HMM*[4] proposed by Salvucci and Anderson [59]. This method is based on a probabilistic finite state machine with one state for the velocity distribution of the saccades and one for the velocity distribution of the fixations. The last category of algorithms, namely *area-based* methods, define static areas of interest to separate the current focus of the test subject. The separation of saccades and fixations inside these areas is usually based on duration thresholds. It is obvious that methods such as *I-AOI*[5] introduced by Salvucci and Goldberg [55] are not suited for dynamic situations with varying AoI.

Despite this first taxonomy, algorithms for eye movement classification can be divided into *threshold-based* and *probabilistic* methods according to Kasneci et al. [56]. Threshold-based methods typically use fixed thresholds, adapted to the respective use case, while probabilistic methods are capable of adapting their thresholds to varying applications and

[1] I-DT = Dispersion-Threshold Identification

[2] I-MST = Minimum Spanning Tree Identification

[3] I-VT = Velocity-Threshold Identification

[4] I-HMM = Hidden Markov Model Identification

[5] I-AOI = Area-of-Interest Identification

changing conditions. An adaptive method called *I-KF*[6] is explained in detail by Sauter et al. in [60] and by Komogortsev and Khan in [61]. The trajectories of the eye movements are predicted by this algorithm using a mathematical model of the human eye. As shown in Table 3.1, only some of these algorithms were applied in driving scenarios. Due to the highly dynamic traffic environments, the inter-individual differences of drivers and camera settings, and the space of movements inside the vehicle, eye movement classification in driving scenarios represents a particularly challenging task. To address this challenge, Tafaj et al. [62] proposed a Bayesian mixture model, in short *BMM*, to learn the thresholds of the algorithm in an online fashion and compared it to the adaptive state of the art algorithm *I-HMM* in a following study [56], revealing a superior classification performance of the *BMM* over the *I-HMM*. The *BMM* considers the Euclidean distances of sequential data points for its classification and is based on the assumption that these distances, describing either a fixation or a saccade, are Gaussian distributed. The parameters of the applied Gaussian Mixture Model (GMM) will be updated with every new classified distance in an online-fashion. Note that if a constant sample time is considered, the distances can be seen as velocities. This approach was expanded to additionally detect smooth pursuits in traffic scenarios [63]. The last approach listed in Table 3.1, called MERCY, is also an extension of the Bayesian mixture model approach and will be introduced in Section 3.2.

[6] I-KF = Kalman-Filter Identification

Study	Name	Spatial Criterion	Threshold	Temporal Criteria	Applied in	Method
Widdel [57], Salvucci and Goldberg [55]	I-DT	Dispersion-based	fixed	duration sensitive, locally adaptive	Laboratory	Moving window
Sauter et al. [60], Komogortsev and Khan [61]	I-KF	Velocity-based	adaptive	-	Laboratory	Kalman Filter
Goldberg and Schryver [58]	I-MST	Dispersion-based	fixed	locally adaptive	Laboratory	Minimum Spanning Tree
Salvucci and Anderson [59]	I-HMM	Velocity-based	adaptive	locally adaptive	Laboratory, Vehicle	Hidden Markov Model
Salvucci and Goldberg [55]	I-AOI	Area-based	fixed	duration sensitive	Laboratory	Fixed target areas
Salvucci and Goldberg [55]	I-VT	Velocity-based	fixed	-	Laboratory	Point-to-point velocities
Tafaj et al. [62]	BMM	Velocity-based	adaptive	locally adaptive	Vehicle	Bayesian Mixture Model
Tafaj et al. [63]	BMM+	Velocity-based	adaptive	locally adaptive	Vehicle	BMM with PCA
Braunagel et al. [12]	MERCY	Velocity-based	adaptive	locally adaptive	Vehicle	Mixture Model

Table 3.1: Chronological overview of existing eye movement classification algorithms.

3.1.2 Necessity of Adaptive Methods

As described in the previous subsection, algorithms for eye movement classification can be separated into threshold-based and probabilistic methods depending on the method used for calculating the necessary parameters of the algorithm. However, which one of these two approaches is the one to favor, or if adaptive methods are necessary at all, are still unresolved topics.

Salvucci and Goldberg [55] suggested that fixed thresholds are usually sufficient for the classification since the velocity profiles are assumed to be physiologically stable. However, determination of the fixed threshold depends on the respective task. As an example, the trajectories of self-paced saccades with a fixed velocity threshold of $15°s^{-1}$ were examined by Erkelens and Vogels [64], while Sen and Megaw used a threshold value of $20°s^{-1}$ to detect effects on saccades while working on visual display units [65]. A summary of further settings for this threshold is given by Rötting in [66]. On the other hand, Kasneci et al. [56] suggested a preference for probabilistic methods in non-automated driving scenarios, since driving scenarios can doubtlessly be considered as far more dynamic environments than lab environments. Therefore, empirically adjusted thresholds are not applicable to these highly dynamic scenarios and the strongly physically- and physiologically-dependent viewing behavior. However, no further references or details on this statement were given.

Besides the literature on automated eye movement classification, there are several studies, typically in the field of psychology, examining the individual viewing behavior during specific tasks. For example, Castelhano and Henderson found inter-individual differences in the saccadic amplitudes during the scanning process of images, while the intra-individual saccadic amplitudes were stable [67]. Moreover, there are a considerable amount of eye movement studies in reading describing differences in the viewing behavior among various types of readers. A comprehensive overview on the above work is given in [68]. All these findings indicate that there are significant individual differences in the eye movement parameters among individuals performing the same task, while the existence of various threshold settings for different tasks suggests a non-negligible task-individual difference.

The scenario of conditionally automated driving exposes further challenges to the eye movement classification algorithms because both task- and inter-individual differences occur at the same time. There are plenty of possible secondary tasks which can be performed in conditionally automated driving scenarios and between which the driver can switch frequently. Examples for possible secondary tasks are reading news, writing emails, watching a movie, or just relaxing and observing the environment. Even the level of automation can change between conditionally automated and non-automated route sections, so that the driver needs to take-over or hand-over the control of the vehicle.

In summary, all these varying task-individual conditions can influence the driver's eye movement behavior. Further, inter-individual differences need to be considered due to the possibility of multiple various drivers per vehicle and drive. Since the eye movement behavior of various drivers can react individually for the different conditions, the task- and inter-individual differences intensify each other. Hence, it can be assumed that conditionally automated driving leads to high variations in drivers' eye movements and makes adaptive methods necessary.

3.2 MERCY-Moving Estimation Classification

As described in the previous section, different adaptive methods for the automated eye movement classification exist which have even been applied to non-automated driving scenarios. In the following, an adaptive state-of-the-art algorithm for recognizing eye movements, namely the *BMM*, shall be described and analyzed in detail. Especially the adaptability to variations of the used data set and other disadvantages concerning the application in a vehicle are examined. Since it is shown that the BMM is not sufficient for use in conditionally automated driving scenarios, an extended version of this algorithm will be introduced and compared to the original one. As shown at the end of this chapter, the adapted algorithm will enable the investigation of the varying eye movement behavior in conditionally automated driving scenarios.

3.2.1 Analyzing the Bayesian Mixture Model Approach

Based on the same assumption as in [62] that the underlying process generating the velocities of fixations and saccades can be described by a GMM, the probability density function $p(||v_i||)$ of the model is given by

$$\begin{aligned} p_f(||v_i||) &= \pi_f \mathcal{N}(||v_i||, \mu_f, \beta_f) \qquad (3.1)\\ p_s(||v_i||) &= \pi_s \mathcal{N}(||v_i||, \mu_s, \beta_s) \qquad (3.2)\\ p(||v_i||) &= p_f(||v_i||) + p_s(||v_i||) \qquad (3.3) \end{aligned}$$

where v_i is the measured velocity between the two sequential data points with the index $i-1$ and i, the parameters μ and β describe the mean and the variance of the Gaussian distribution, π describes the mixture parameter, and the indices f and s denote the components of the fixations or the saccades. The norm $||.||$ represents the Euclidean distance. The classification process can be seen as the determination of the intersection δ of the two probability density functions p_f and p_s and consequently boils down to the estimation of the means, variances, and mixture components denoted by the parameter set

$$\Theta_k = \{\mu_k, \beta_k, \pi_k\} \quad where \quad k \in \{f, s\}. \qquad (3.4)$$

This intersection point δ represents the adaptive classification threshold used for detecting saccades and fixations. Therefore, the terms *intersection point* and *threshold* are used interchangeably for the parameter δ in the following. An artificially generated example of a GMM is shown for illustration in Figure 3.1.

To determine the parameter sets Θ_f and Θ_s, Tafaj et al. [62] used Variational Message Passing (VMP) as implemented by Infer.NET[7] [69]. Since VMP is an advanced approximation technique for applying variational inference to Bayesian Networks, the time required as well as the complexity for implementing such a framework enable running it online on

[7] http://research.microsoft.com/en-us/um/cambridge/projects/infernet/

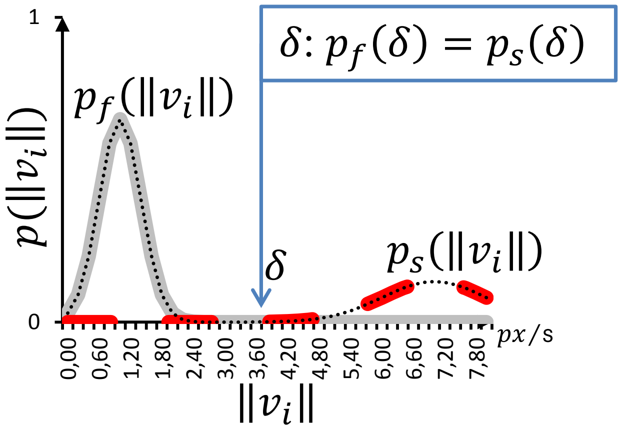


Figure 3.1: The three probability density functions p_f (grey), p_s (red), and p (dotted black).

common rapid control prototyping (RCP)[8] and hardware in the loop (HiL)[9] tools in the vehicle are still enormous. Moreover, the VMP algorithm is realized as an iterative method converging in terms of a lower bound [69]. For this iterative approach, it cannot be determined a priori how many iterations need to be performed, which can be problematic in terms of real-time applications.

3.2.2 Analyzing MERCY

This subsection introduces MERCY, a novel approach for an improved estimation of the parameters of GMMs and suitable for implementation on common RCP and HiL tools in the vehicle. The architecture of this approach is illustrated in Figure 3.2 and can be separated into three iterative steps: estimation, updating, and classification. These steps are performed in each iteration, requiring the parameter sets Θ_f and Θ_s of the previous round and the current measured velocity.

Classification

Classification is performed in the same way as in the BMM algorithm by comparing the current velocity $||v_i||$ to the intersection point δ. If the velocity is smaller than the intersection point, i.e. it lies on the left side of the intersection, it will be marked as a fixation or otherwise as a saccade. After the classification, the algorithm is able to update one of the two distributions depending on the belonging of the current velocity.

[8] Iterative method for designing and testing control strategies.

[9] Test bench including the embedded system and the replication of a realistic environment for generating the input of the system.

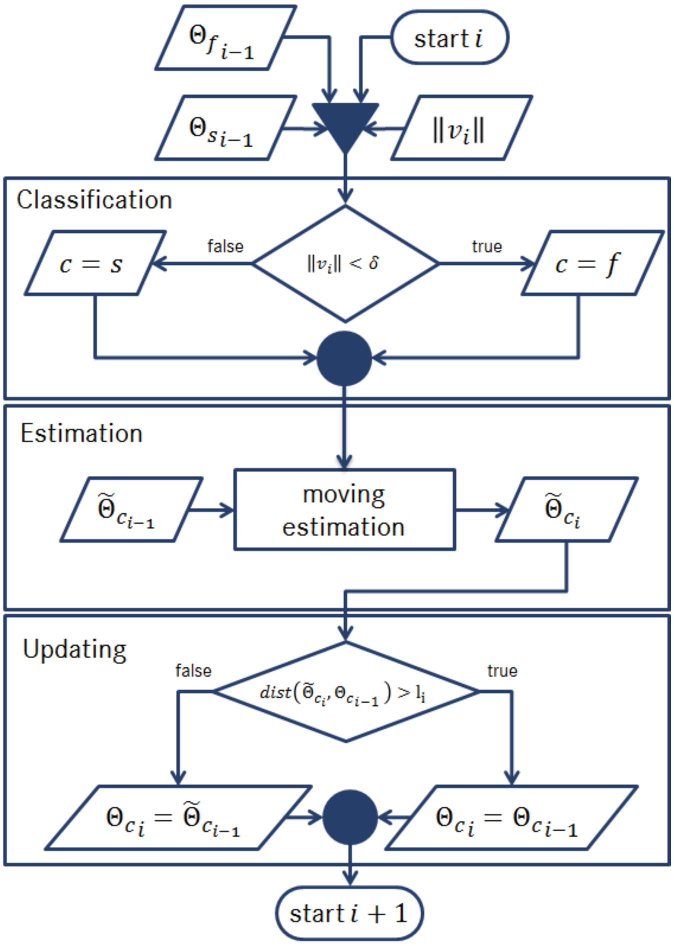

$\Theta_{f_{i-1}}$
start i
$\Theta_{s_{i-1}}$
$\|v_i\|$
Classification
false
$\|v_i\| < \delta$
true
$c = s$
$c = f$
Estimation
$\widetilde{\Theta}_{c_{i-1}}$
moving estimation
$\widetilde{\Theta}_{c_i}$
Updating
false
$dist(\widetilde{\Theta}_{c_i}, \Theta_{c_{i-1}}) > l_i$
true
$\Theta_{c_i} = \widetilde{\Theta}_{c_{i-1}}$
$\Theta_{c_i} = \Theta_{c_{i-1}}$
start $i+1$

Figure 3.2: Architecture of the novel algorithm MERCY.

Estimation

The main idea behind MERCY is estimating the parameter sets Θ_f and Θ_s by means of sample mean and sample variance. These estimations of the means, variances, and mixture components can easily be reformulated into a recursive form. Further, to prevent the estimation from converging and that new data samples will be considered with decreasing weight, the recursive formulas can be provided with a weighting factor ω which can be interpreted as the size of a moving window. Choosing a small ω leads to an extremely dynamic behavior of the estimation, but increases the influence of outliners on the estimation. On the other hand, choosing a large ω results in idle behavior which adapts slowly to the changing conditions. Given the simplifying assumption that the velocities $v_1, v_2, \ldots$ are realizations of the random variable $\mathcal{Z}$ generated by an independent and identically distributed process, the recursive equation for the weighted sample mean is defined as

$$\mu_{k_{n+1}} = \frac{\omega\,\mu_{k_n} + v_{n+1}}{\omega + 1} \tag{3.5}$$

and the recursive equation of the weighted sample variance is defined as

$$\beta_{k_{n+1}} = \frac{(\omega - 1)\beta_{k_n} + (v_{n+1} - E[V_{n+1}])^2}{\omega} \tag{3.6}$$

with $E[V_{n+1}]$ representing the expectation of the set of the last $n+1$ sequential velocities given by V_{n+1}. Similar to the previous chapter, the index $k = \{f, s\}$ denotes the components of the fixations or the saccades whereas the second index n describes the sample point. The expectation value $E[V_{n+1}]$ in equation (3.6) can be replaced by the sample mean of equation (3.5) to

$$\beta_{k_{n+1}} = \frac{(\omega - 1)\beta_{k_n} + (v_{n+1} - \mu_{k_{n+1}})^2}{\omega}. \tag{3.7}$$

Note that the estimation of the variance depends on the estimation of the mean of the same round. The estimation of the mixture components π_f and π_s, which describe the ratio between the number of data samples classified as fixations and saccades, is realized by means of a weighted counter given by

$$\pi_{k_{n+1}} = \frac{\omega\,\pi_{k_n} + 1}{\omega + 1}. \tag{3.8}$$

In comparison to the estimation of the sample mean and sample variance, both parameters π_k can be updated in every round of the algorithm independent of the classification result.

Updating

As shown in Figure 3.2, there are two pairs of parameter sets Θ_k and $\tilde{\Theta}_k$. While the parameter set Θ_k describes the actual parameters used for the classification, $\tilde{\Theta}_k$ depicts the currently estimated parameters of the GMM with regard to the new data samples. If these parameter sets diverge by more than a pre-defined threshold l, the currently estimated parameters $\tilde{\Theta}_k$ will be used as new parameter set Θ_k for the classification in the next round. As long as the threshold is set to $l = 0$, the actual model parameters will be updated with every new data sample, which is generally the proper approach. Nevertheless, this separation into two parameter sets was considered as a possible additional analysis of the task- and inter-individual differences. Since the whole algorithm has a constant complexity $O(1)$, this method is suitable for most real-time applications.

The reliability of MERCY depends mainly on the choice of the initial parameters of the GMM. Hence, these initial parameters should at least be in the same range as the average parameters over as many drivers and situations as possible. Therefore, random segments of a pre-defined size were extracted from randomly chosen simulator drives of the Pre-Study NEBAF described in Subsection 2.4.3 and used as input for the Expectation

Maximization algorithm[10]. The estimated parameters by means of the Expectation Maximization algorithm were averaged, resulting in the initial values $\Theta_{f,init} = \{0.55, 1.13, 0.90\}$ and $\Theta_{s,init} = \{30.09, 3792.20, 0.10\}$ applied for some of the later evaluations. Weighting factor $\omega = 10$ was used throughout this study so that the algorithm could react to current changes in the eye movement behavior within a short period of time and, at the same time, not generate high-frequency oscillations. Moreover, the threshold l was set to $l = 0$ so that the parameters were updated in every iteration.

3.2.3 Comparative Evaluation

At first, artificially generated data is used to evaluate the ability of the BMM to adapt its parameters to frequently changing eye movement behavior. The reason for this approach is to provide a Ground Truth of the GMM and its parameter sets, which facilitates a simulation of the frequent changes of the underlying process and the evaluation in total. The MATLAB class *gmdistribution*[11] was used to create the GMM because it can generate random numbers of the specified mixture model. In total, 25000 random data samples were generated with different pre-defined parameter sets for a first evaluation. After 5000 samples, the GMM's parameter set was changed for the first time and another 5000 data samples were generated. In Figure 3.3, this procedure was repeated four times, before 5000 random data samples were generated by a step-by-step changing model, resulting in a continuously decreasing threshold at the end of the figure. Table 3.2 specifies which parameters were used and varied for the different intervals, each containing the 5000 samples. The BMM was trained with the first 1000 data samples before starting the online adaption, which explains the gap at the beginning of the BMM plot in Figure 3.3. The initial parameters as well as the variations of these values in Table 3.2 were chosen to provide a meaningful GMM according to preliminary studies while still providing distinctly separable data for a moderate classification task.

As shown in Figure 3.3, the BMM based on the velocity distributions adapts poorly to the frequently changing GMM. Among other things, it shows the Ground Truth threshold of the artificial GMM and the threshold of the estimated GMM of the BMM. After the training phase, the estimated threshold of the BMM differs from the actual threshold, but is slowly approaching it. The reason for this slow behavior is that the generated data samples are weighted lower the later they are given to the algorithm. Hence, shortly after the training phase, new data samples already have little to no effect on the parameters of the BMM. Consequently, for significantly emerging differences such as at sample $15,000$ in the fourth interval, the BMM is overwhelmed by the adaption process. In subsequent sections it will be shown that the individual differences due to performing different tasks are significantly larger than for the artificially generated data at this point. Hence, an even worse estimation of the mixture model in case of conditionally automated driving data is

[10] The Expectation Maximization algorithm is an iterative method for estimating the most probable parameters of a statistical model.

[11] http://de.mathworks.com/help/stats/index.html

Parameter set of the x-th interval					
interval	1	2	3	4	5
μ_1	1	1	1	1.1	$1.1 - i\frac{1}{e^9}$
μ_2	200	220	210	207	207
β_1	0.1	0.1	0.14	0.33	0.33
β_2	400	400	400	404	$404 - i\frac{1}{e^3}$
π_1	0.8	0.5	0.6	0.7	0.7
π_2	0.2	0.5	0.4	0.3	0.3

Table 3.2: Parameter sets of the different 5000 samples large intervals shown in Figure 3.3. The varied values between two sequential steps are highlighted by a blue background. The parameter i in the last column represents the i-th iteration, since these values were varied for every iteration.

expected.

MERCY is applied to the same artificial data samples as the BMM and the estimation is plotted as a light gray line in Figure 3.3. Although the initialization values were chosen with an offset of 0.5 in the means and variances, resulting in a starting position of the estimated intersection point at $4\,px/s$, the algorithm adapts as fast as the BMM to the artificial model. However, MERCY is more accurate than the BMM up to iteration $15,000$. In contrast to the BMM, MERCY is still able to detect and adapt to the changing distribution in the fourth interval, but the error between the actual threshold and the estimation increases slightly, due to the lack of a sufficient number of data samples. The performance of MERCY exceeds the performance of the BMM for larger steps in threshold δ, and MERCY adapts appropriately even in the fifth interval with the continuously decreasing intersection.

A large-scale data set of half a million data samples, generated by randomly changing parameters of the artificial GMM was created in the same way as in the exemplary plot of Figure 3.3. The parameters were varied randomly every $10,000$ samples so that every parameter, e.g. the mean μ_f, was set to a value of the interval defined by the initial value and the radius, e.g. $[\mu_f - \mu_f/2, \mu_f + \mu_f/2]$. Furthermore, every $50,000$ samples, MERCY was reset to the initial values and the parameter of the BMM were determined by an additional training phase. To compare the performance of both algorithms, the absolute error between the actual intersection point and the estimated points was calculated. The result is shown as a stacked bar diagram in Figure 3.4. For the plot, the intervals of the training phases of the BMM and all absolute errors smaller than $0.1\,px/s$ were not considered. In addition, one round of $50,000$ samples was discarded because the BMM was not able to calculate a meaningful initialization of the model in the training phase. The calculated error was separated into three different error classes, dividing them into small errors $\leq 1.2\,px/s$, medium size errors between $1.2\,px/s < x \leq 2.4\,px/s$, and the class of the large errors with $2.4\,px/s < x$. The stacked bar of MERCY shows no errors for the large errors class, since

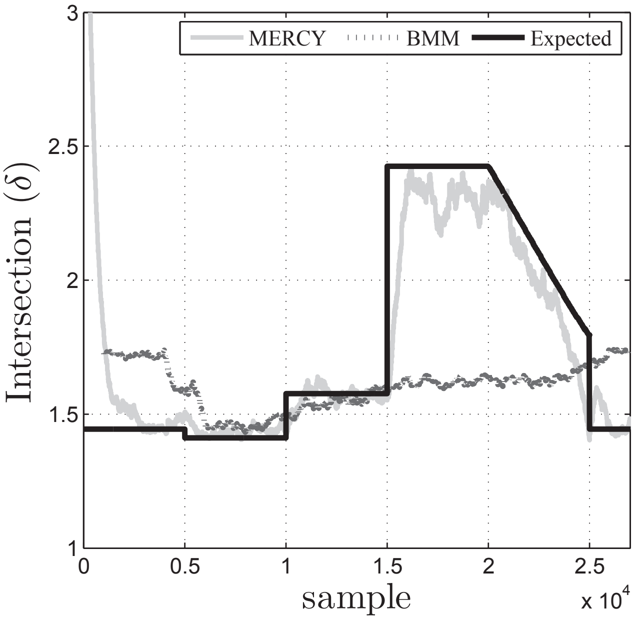


Figure 3.3: The three plots show the artificially generated threshold (black solid line), the estimated threshold by the BMM (dotted gray line), and MERCY (solid light gray line).

there are too few to be visible in the plot. There are $25,000$ errors smaller than $0.1\,px/s$ resulting in a decreased bar height compared to the bar of the BMM. The right bar can be coarsely divided into one quarter of medium size errors and three quarters of small size errors. In contrast, the bar of the BMM can be divided into three nearly equal stacks of the different error classes. As suggested by the example in Figure 3.3, MERCY adapts considerably better to the given data samples than the BMM, providing fewer and smaller errors in terms of the intersection point. All errors of every parameter of Θ_f and Θ_s affect the estimation of the intersection point, which therefore can be seen as the worst-case scenario for the estimation.

However, a thorough comparison of MERCY with the original BMM approach and other off-the-shelf algorithms has to contain an evaluation of the detection performance based on real-world data sets. Hence, an evaluation of MERCY in comparison to the BMM and the dispersion-based algorithm *I-DT* was performed regarding their capability to distinguish between fixation and saccade points. In total, eight data sets of six different subjects participating in the Pre-Study NEBAF and performing the secondary tasks described in Subsection 2.4.3 were used. These data sets consisting of 6623 fixation points and 1384 saccade points were manually labelled by two raters. The duration and dispersion thresholds of the *I-DT* were set to the fixed values of $100\,ms$ and $15\,px$ in terms of the unit of the eye camera. For each subject, the BMM was trained with 1000 unlabelled data samples before the actual evaluation. The initial values of MERCY were set to the estimated values $\Theta_{f,init} = \{0.55, 1.13, 0.90\}$ and $\Theta_{s,init} = \{30.09, 3792.20, 0.10\}$ with a weighting factor of

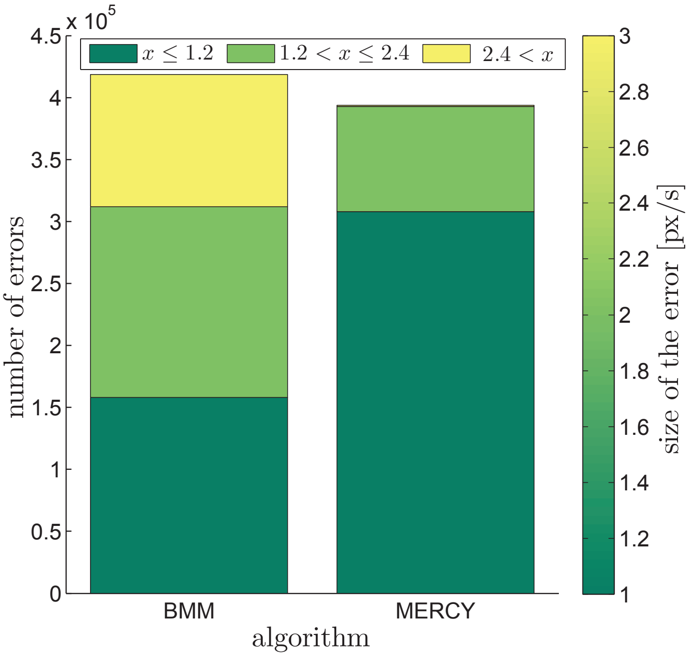


Figure 3.4: Two stacked bars illustrate the absolute error between each algorithm and the actual intersection point of the artificial GMM. The three stacks per bar represent three classes of error sizes.

Algorithms	Recall	Precision	F1 score
I-DT	0.73	0.66	0.69
BMM	0.86	0.67	0.75
MERCY	0.91	0.75	0.82

Table 3.3: Summary of the classification results of the algorithms I-DT, BMM, and MERCY based on labelled real-world data sets.

$\omega = 10$ and a threshold $l = 0$ proposed in Subsection 3.2.2.
As shown in Table 3.3, MERCY achieved the best results in classifying the data points to the correct eye movement type for all three metrics: recall[12], precision[13], and F1 score[14] of the applied algorithms. The BMM showed a high recall value, since it is sensitive to even small point-to-point velocities. However, this sensitivity leads to an increased false negative rate and, therefore, to the low precision on the labelled data set. The threshold-based algorithm *I-DT* showed the lowest results for all three metrics, indicating the disadvantage of the fixed threshold versus the adaptive ones.

[12] Recall = TP/(TP+FN)

[13] Precision = TP/(TP+FP)

[14] F1 score = 2TP/(2TP+FP+FN)

In summary, despite the simple implementation, the introduced approach provides an improved adaptability for the classification of eye movements during frequently changing viewing behavior and is suited for real-time applications due to the complexity on the order of $O(1)$. Further, MERCY outperformed the BMM and the I-DT in classifying saccades and fixations based on labelled real-world data sets.

3.3 Eye Movement Behavior in Conditionally Automated Driving Scenarios

In the following evaluation only 74[15] of the initial 85 experiments of the Pre-Study NEBAF described in Subsection 2.4.3 could be used due to missing signals from the eye tracker for six subjects and erroneous simulations such as traffic freezes for five subjects. In total, the eye movement data set included 35.5 hours of recorded eye tracking data separated into 1.5 hours of manual and 34 hours of conditionally automated driving. First, accumulated eye movement behavior of the experimental versus the control group was investigated for significant differences. For this purpose, MERCY was applied to the eye-tracking data of each driver, since the adaptability of this method proved to be convenient for describing mixture models and their variations. Again, the initial values were set to $\Theta_{f,init} = \{0.55, 1.13, 0.90\}$ and $\Theta_{s,init} = \{30.09, 3792.20, 0.10\}$ with a weighting factor of $\omega = 10$ and a threshold $l = 0$.

First evidence of an existing difference in the eye movement behavior between conditionally automated driving scenarios with and without performing secondary tasks is provided simply by looking at two examples of the curve shape of the intersection δ in Figure 3.5. While the blue solid plot, representing the intersection point of one of the idle drivers, appears to be stable and shows only high-frequency noise, the red dashed plot of one of the drivers performing the secondary tasks shows huge drifts throughout the whole experiment. These drifts could be the result of the task-individual eye movement behavior, which would be a strong evidence for the authors' hypothesis that frequent changes in the performing task lead to a significantly varying eye movement behavior. In addition, the huge differences of up to 90 px/s in the amplitudes as well as the steep gradients of the shown drifts require an even higher adaptability of eye movement classification algorithms than the artificially generated data in Subsection 3.2.3. The vertical offset of the two curves can be attributed to inter-individual differences due to variations of the individual viewing behavior or the settings of the measuring system, e.g. decreased distance of the camera to the eye.

To analyze the intersection behavior over all subjects, Figure 3.6 shows the boxplots of δ, averaged over the whole test duration of every subject. For the plot, possible outliers were removed by considering only the inner 95% of the data samples. Applying the one-

[15] 41 males/33 females, mean age of 39 years (range 20-60, SD=10)

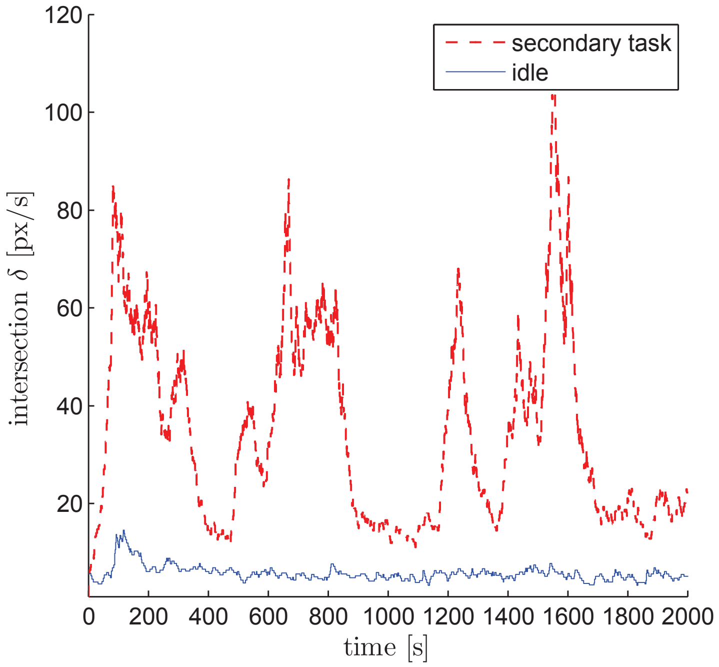


Figure 3.5: Exemplary plots for the behavior of the intersection point of a driver of the experimental and control group.

sample Kolmogorov[16]-Smirnov[17] test to the estimated data of the intersection point, it was indicated that the data is not normally distributed. The difference of the eye movement behavior between the experimental group and the control group can be seen straightaway in Figure 3.6, since there is no overlapping of the interquartile ranges, including median, first and third quartile, of the two boxplots. This first impression is underpinned by the Wilcoxon[18] rank-sum test and the Hedges'[19] g measure, implying that the difference is significant ($p = 0.002$, $z = 2.99$) and of practical relevance ($g = 0.711$). Despite the increased value in the location parameters, the left boxplot shows an increased interquartile and whisker range. These findings illustrate the significant difference in the estimated intersection point between both groups and therefore suggest that the variations in eye movement behavior are considerably greater for drivers performing secondary tasks than

[16] Andrei Nikolajewitsch Kolmogorow, * 25. April 1903 in Tambow, Russland, † 20. October 1987 in Moscow, mathematician who made significant contributions to probability theory, topology, and to other scientifical areas.

[17] Nikolai Wassiljewitsch Smirnow, * 17.October 1900 in Moscow, † 2.June 1966 in Moscow, statistician who researched nonparametric statistics.

[18] Frank Wilcoxon, * 2. September 1892 in County Cork, Ireland, † 18. November 1965 in Tallahassee, Florida, was an american chemical scientist and statistician.

[19] Larry Hedges, Professor for Statistics, Education, and Social Policy at the Institute for Policy Research, Northwestern University.

for drivers without any tasks.

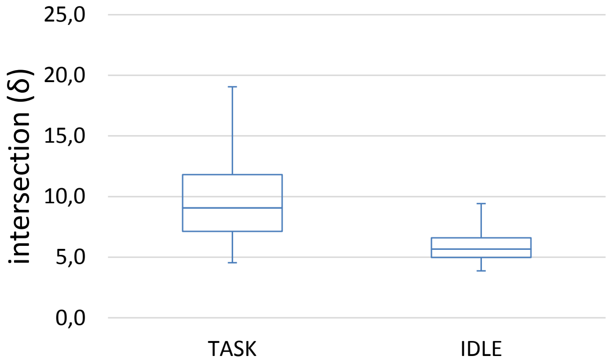


Figure 3.6: Boxplots of the averaged estimated intersection point δ of MERCY while performing secondary tasks versus being idle. The boxplots show the inner 95% of the data, excluding in this way the lowest and the largest 2.5% of the data due to outliers.

To identify the parameters of the estimated GMM, which vary the most during the conditionally automated driving scenario and which differ between the idle and busy driver, Table 3.4 compares the means, medians, minimums, maximums, and variances of the parameter sets Θ_f and Θ_s of the experimental and control group.

		mean	med	var	min	max
Task	μ_f	0.73	0.69	0.08	0.26	1.96
	μ_s	55.42	52.35	337.87	21.71	89.04
	β_f	2.26	1.96	2.51	0.55	9.73
	β_s	3604	3733	241763	1763	3999
	π_f	0.91	0.92	0.001	0.73	0.98
	π_s	0.09	0.08	0.001	0.02	0.27
		mean	med	var	min	max
Idle	μ_f	0.53	0.50	0.03	0.23	1.46
	μ_s	50.35	46.41	262.7	21.47	80.60
	β_f	1.32	1.15	0.65	0.49	6.00
	β_s	3622	3761	179901	1825	3999
	π_f	0.91	0.92	0.001	0.71	0.97
	π_s	0.09	0.08	0.001	0.03	0.29

Table 3.4: Statistical values of the estimated GMM divided into mean, median (med), variance (var), minimum (min), and maximum (max).

It can be seen that parameters μ_f, π_f and π_s for both groups of subjects have such low variances that these parameters probably do not require learning and adapting to them at all. Especially the a priori probabilities π_f and π_s imply a constant ratio of $1/9$ between saccades and fixations over the whole experiment and all statistical measures are nearly identical for both groups. The average velocity of fixations μ_f is not exactly zero as expected, due to measurement inaccuracies or smaller eye movements such as the nystagmus[20]. Nevertheless, as long as such "disturbances" are kept as small as possible, there will be no significant variation in this parameter. The size of the relative variances as well as the ranges from the minimum to the maximum value of the remaining parameters μ_s, β_f, and β_s indicate that these values vary the most overall the measured data. Note that due to the flat and wide distribution of the saccades, the influence of μ_s on the intersection point and hence on the classification is low. In summary, for the given assumption of a GMM describing the process of generating saccades and fixations, it would be sufficient to learn only parameters β_s and β_f, describing the variance of the distribution of the saccades or the fixations, since the remaining parameters of the mixture model can be considered as constant or their influence on the classification performance is vanishingly low. If the values of Table 3.4 are compared between the control and experimental group, an increased variance is observed for secondary tasks performed for the intersection point. This finding confirms the hypothesis suggesting high variations in eye movement behavior due to task-individual differences as analogous to the evaluation of the estimated parameter δ above.

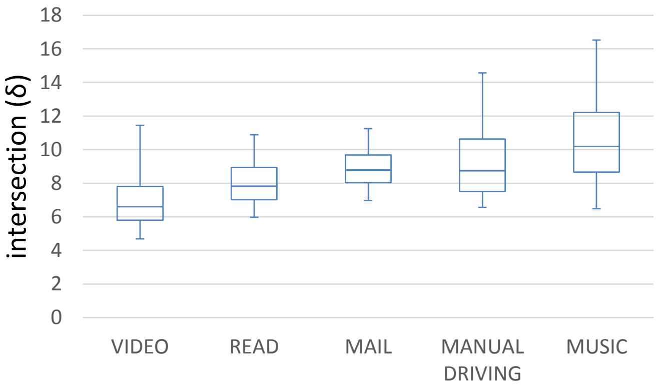


Figure 3.7: Quantitative comparison of the behavior of the estimated intersection point during the different secondary tasks and while driving manually over all subjects.

To explain which secondary tasks cause the variation in the eye movement behavior during conditionally automated driving, Figure 3.7 shows a boxplot of the estimated intersection point of all performed secondary tasks and of the manual driving sections. The interquartile range of the boxplots of the tasks *video*, *mail*, and *read* are similar to the range of the idle task, but with an increased average of the estimated intersection point. These small variances probably result from the fact that all three tasks were performed on the touch screen built in the cabin. Thus, most eye movements were performed in a narrow field of view. Obviously, this cannot be the sole explanation of the increased variations in the eye movement behavior during the performing of the secondary tasks. In contrast, the *music* task reveals a larger variation of the eye tracking data than the idle task in Figure 3.6, although a similar viewing behavior of both tasks is expected. A possible explanation for

[20] Rhythmic, oscillating, and involuntary movements of the eyeball [70].

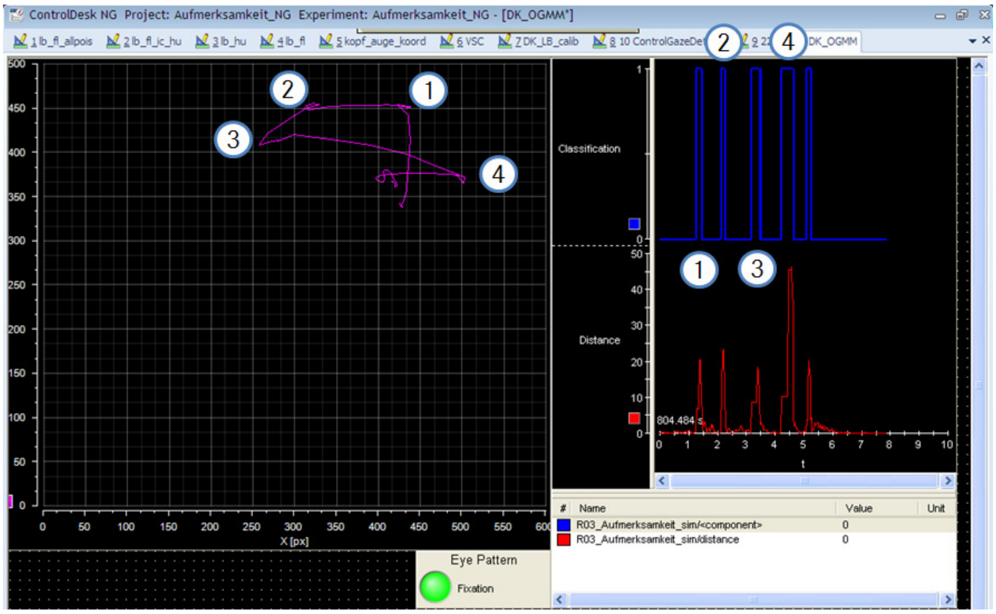
this larger variation could be the gazes of the driver on the touch screen, since the display was not turned off during the *music* task and, therefore, could still attract the attention of the driver. Another explanation could be the more active scanning by the driver of the environment between the usual tasks performed on the touch screen which force the driver to focus their attention on the display and not to observe the environment at length. An interesting point to mention is the high variation of the intersection point of the manual driving scenarios. This result indicates that for non-automated driving scenarios on a typical german highway, significant variations in the viewing behavior occur which need to be taken into account for a robust eye movement classification.

In summary, the tasks can be separated into two groups regarding their variation of intersection δ: one group comprising the *music* and *manual driving* tasks and showing large variations, and a second group containing the remaining tasks *read*, *video*, and *mail* and depicting small variations. Since these two groups alternate frequently in conditionally automated driving scenarios, the eye movement behavior switches between tasks with small and larger variations, leading to the higher variation of the viewing behavior while performing secondary tasks compared to when idle. Hence, the provided evaluation proves the necessity for adaptive thresholds for algorithms for eye movement classification in conditionally automated scenarios.

3.4 Applying MERCY online in the Vehicle

As mentioned in the previous sections, MERCY is suited for real-time and hardware-in-the-loop tools applied in testing vehicles. Especially the application of the moving estimation based on the recursive formula in (3.5), (3.6), (3.8) enables short runtimes and a low memory consumption. Furthermore, only the results of the previous iteration are applied, rendering large memory buffers unnecessary. For demonstration purposes, MERCY was implemented as described in Section 3.2 in Simulink and compiled to run on the MABX of the testing vehicle described in Subsection 2.4.2. Further, a GUI was designed for visualizing the classification output as well as the raw eye-tracking input of MERCY to highlight the benefit of an adaptive threshold.

Figure 3.8 shows the GUI during an on-going measurement. The coordinate system on the left of the GUI visualizes the raw eye-tracking data in original pixel coordinates of the camera. Therefore, the coordinate system has a resolution of 384×288 pixels. The classification result is shown by the binary blue signal in the upper right corner, where the value is set to one if a saccade is detected. In addition, a light is shown with the corresponding name of the detected eye movement pattern at the bottom of the GUI. The light is switched to red for saccades and to green for fixations. Below the classification area, the Euclidean distance of sequential data points is shown as a red signal. Further, the discrete values of the classification signal (blue signal) and the signal describing the Euclidean distance (red signal) are displayed in the lower right area. As shown in the subsequent Figures 3.9 and 3.10, the color of the signals is adjustable by the user. The signals are shown for a window of ten seconds before vanishing. Saccades with such

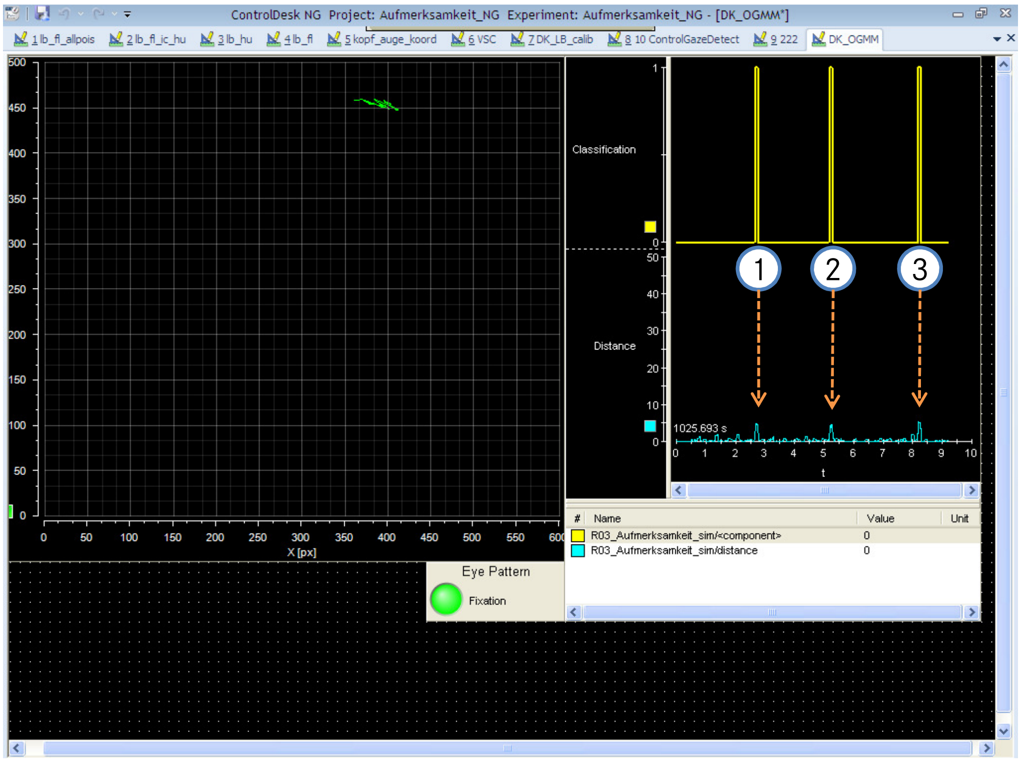


#	Name	Value	Unit
🟦	R03_Aufmerksamkeit_sim/<component>	0	
🟥	R03_Aufmerksamkeit_sim/distance	0	

Figure 3.8: Four saccades were performed in the test vehicle and visualized on the described GUI. The endpoints of the saccades are consecutively numbered from 1) to 4): 1) Saccade from the center console to the interior mirror, 2) to the windshield, 3) to the left exterior mirror and 4) to the right exterior mirror.

large amplitudes as the performed saccades in Figure 3.8 to the interior mirror, to the windshield, and to the left and right exterior mirrors shown in Figure 3.8, are easily detected by MERCY online and are highlighted by numbers.

However, these large saccades can also be easily detected by means of algorithms based on fixed thresholds. To illustrate the benefit of MERCY online, two additional typical eye movement behaviors for conditionally automated driving scenarios were performed. In Figure 3.9, the test subject was reading a short message of three lines on the central display of the vehicle. On the area of the pixel coordinate system, the typical serrated eye movement pattern for reading sequences is plotted. It can be seen that the Euclidean distances are quite small, even the slightly larger saccades to the left at the end of the line. Algorithms with fixed thresholds experience difficulties with such small saccades since these eye movements can only be detected for small thresholds, making them vulnerable to measurement noise. The benefit of adaptive approaches lies in the ability to decrease the threshold only during periods where small saccades are performed and increase the threshold again afterwards. In the example shown in Figure 3.9, the test subject was reading a text for about ten seconds before this screenshot was taken. Due to the small amplitudes of the performed eye movements, the Euclidean distance only shows small peaks. Thus, the saccades at the end of the lines are classified correctly due to the adaption

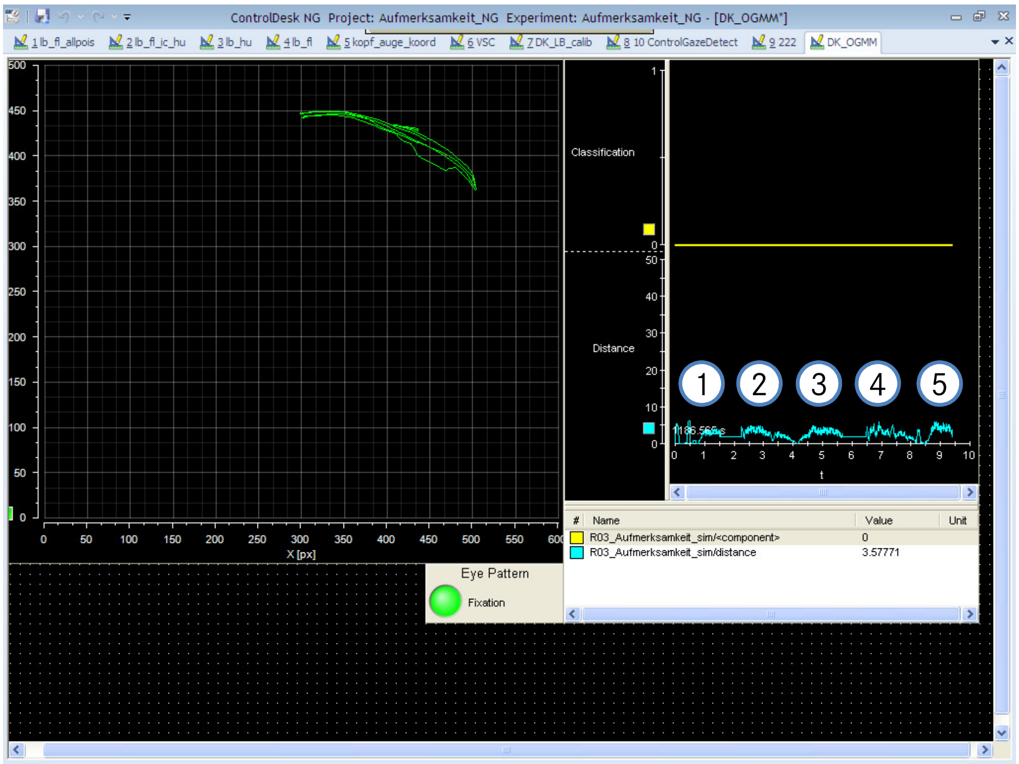


#	Name	Value	Unit
🟨	R03_Aufmerksamkeit_sim/<component>	0	
🟦	R03_Aufmerksamkeit_sim/distance	0	

Figure 3.9: For this figure, the test subject was recorded during a reading sequence of ten seconds. In this time, the algorithm adapted to the corresponding eye movement pattern, including small saccades. The three saccades at the end of the line were detected correctly and are highlighted in the GUI.

of the mixture model and the shift of the intersection point to smaller values. Obviously, this adaption is only possible if the mixture model obtains an adequate interval for the adaption process.

Although these saccades with small amplitudes are detected by MERCY online in the vehicle, the classification algorithm is robust against variations due to eye movement patterns with similar amplitudes. For example, Figure 3.10 shows the eye-tracking data and the output of MERCY for smooth pursuits. The test subject was focusing on an object which was continuously moved from the left to the right in front of the standing vehicle. Although the distances reach similar values as the eye movements in Figure 3.9, no saccades are detected even after several minutes of performing smooth pursuits. In contrast to the example of Figure 3.9, smooth pursuits usually do not create sequential Euclidean distances with increased amplitudes which are classified as saccades and, therefore, the threshold of the GMM does not decrease even for longer intervals of performing smooth pursuits. Eye movement classification algorithms with a small fixed threshold, such as the I-DT, would usually falsely classify saccades due to the large eye movements. In summary, MERCY



#	Name	Value	Unit
🟨	R03_Aufmerksamkeit_sim/<component>	0	
🟦	R03_Aufmerksamkeit_sim/distance	3.57771	

Figure 3.10: Multiple smooth pursuits were recorded where the test subject was focusing on an object moving five times from the left to the right. For this measurement, no saccades were falsely detected by MERCY.

could be easily implemented on the hardware of the testing vehicle and its general viability was successfully validated through online vehicle test cases.

3.5 Summary

In this chapter, a new method named MERCY for eye movement classification was introduced. MERCY can be seen as an extension of the Bayesian Mixture Model approach proposed by Tafaj et al. in [62] which was already applied in dynamic driving scenarios. However, for implementations on common RCP and HiL tools in the vehicle and for applications in conditionally automated driving scenarios, the Bayesian Mixture Model is too sophisticated and lacks the adaptability to the individual gaze behavior. In contrast, it was shown that MERCY exceeds state-of-the-art approaches including the Bayesian Mixture Model for eye movement classification in both classification performance and general adaptability based on half a million randomly generated data samples and a thorough conditionally automated driving study. Despite excellent classification performance, MERCY is based on simple mathematics and therefore is easy to implement. For demonstration purposes and to verify MERCY online in the vehicle, it was implemented in Simulink including a graphical user interface and transferred to a testing vehicle. Due to the high adaptability of MERCY, the task-individual difference was shown to be significant between the viewing behavior of subjects performing secondary tasks and idle subjects, both driving in a conditionally automated setting. The findings suggest that the eye movement behavior during changing tasks varies constantly and therefore the threshold for the classification between saccades and fixations is varies, too. This indicates the necessity for adaptive thresholds for this task, which until now was an unresolved topic in the field of eye movement classification.

Since MERCY uses sample mean and variance estimators, a reliable estimation of the variance first requires a good estimation of the sample mean. That means that in case of sudden changes in eye movement behavior, the variance is estimated insufficiently as long as the mean has not approximated the actual mean, causing an overshoot of the intersection parameter. The error of the estimation of the sample mean impacts the estimation of the variance quadratically. A possible solution could be a correction function depending on the gradient of the sample mean. Moreover, MERCY is updates only the parameter set Θ_f or Θ_s of the estimated GMM of the current classification result. In case of a large overlap of the two Gaussian distributions, e.g. due to poor initialization values, the incorrect parameters are often updated. Since the total error of the falsely classified data samples can be estimated, this error should be considered in the estimation of the parameters of the model in terms of error minimization. In this way, both parameter sets Θ_f and Θ_s can be updated in every iteration.

In the following chapters, MERCY will be applied as the subsystem for some of the proposed algorithms for Eyes-on-Road detection and driver-activity recognition.

4 Eyes-on-Road - Definition and Application

While driving in an automated setting, drivers have the opportunity to take their eyes off the road, e.g. to perform secondary tasks, since there is no need for a detailed monitoring of the traffic environment. To determine whether the driver is focusing on the road, various systems were introduced in the literature and are even available in series vehicles. These systems and algorithms are subsumed by the *Eyes-on-Road* concept. When performing secondary tasks in conditionally automated driving scenarios, many drivers tend to gaze towards the street, the instrument cluster, or the vehicle's mirrors. These gazes allow a reorientation of the driver in terms of the current traffic situation and, therefore, have an impact on the take-over quality in take-over situations. Especially the detection of these typically short Eyes-on-Road gazes is challenging in real-world traffic environments due to various lighting conditions or not visible eyes due to large head angles. These challenges cannot be solved solely by improved hardware and computer vision algorithms at the moment. Hence, in this chapter novel algorithms based on given eye- and/or head-tracking signals will be introduced to improve the Eyes-on-Road detection. After a phenomenological description of Eyes-on-Road gazes and a summary of existing methods for Eyes-on-Road Detection in the first two Sections 4.1 and 4.2 of this chapter, it will be shown in Section 4.3 that a relative gaze direction can enable a highly accurate Eyes-on-Road detection without any kind of calibration. For the case of missing eye gaze signals, e.g. due to large head angles or camera systems without eye-tracking, an architecture for detecting Eyes-on-Road gazes solely on the head movements is introduced in Section 4.4. As in the previous chapter, all algorithms will be analyzed with regard to their applicability in a real-world testing vehicle with a close-to-production camera system at the end of this chapter.

4.1 Visual Attention and Eyes-on-Road Gazes

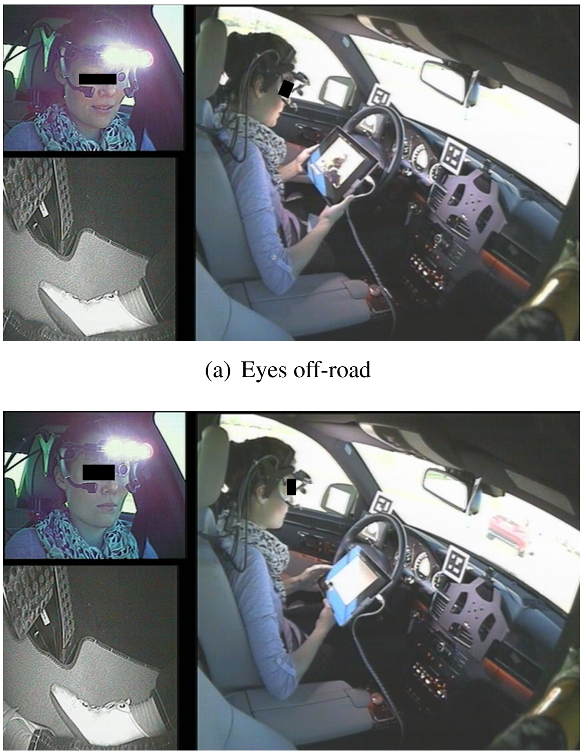
The majority of sensory perception in daily life, in detail about 80%, is received over the visual sensory channel, which corresponds to a transmission rate of about 6.5 MB/s [71]. It is hardly surprising that according to Sivak this statement can be transferred to driving situations [72]. Hence, visual distraction of the driver is considered to be one of the most critical conditions and frequent reasons for near- and actual crashes in the traffic environment [73]. Multiple studies were conducted to examine the correlation between visual distraction and driving performance, e.g. Wierwille and Tijerina in [74] or Jamson and Merat in [75]. Common reasons for visual distraction in non-automated vehicles are secondary tasks or stimuli from the environment. For example, Greenberg et al. [76] performed a thorough investigation of the impact of various versions of using a mobile phone while driving, such as hands-free and hand-held phone dialing. The study showed that the rate of missed events in front of the vehicle is similar for hands-free and hand-held devices due to the increased visual demand of the secondary tasks. In another study, Wallace [77] showed that stimuli from outside of the vehicle can be a dangerous threat to the safety of the driver and other traffic participants. Especially advertisements and signs at junctions or on long monotonous roads may distract the driver significantly.

At this point the question arises, how visual distraction or attention can be assessed in the traffic environment. By fixating any location with the fovea centralis of the eyes, drivers may focus their visual attention to the chosen target. However, it should be noted that the eyes are not able to process all the information inside the visual field of view at once. Instead, different properties of the same target, such as the color or shape, can be observed sequentially [78]. By performing eye movements, such as saccades, the visual focus can be shifted to other targets. Hence, the gaze behavior, in detail the location and the duration of a fixation as well as the performed movements of the fovea, can be associated with visual attention or visual distraction, respectively. Many studies have already proven this close relation between gaze behavior and visual attention, e.g. refer to Frischen [79] for an overview of this topic in daily life social interactions or Chapman and Underwood [80] with regard to the traffic environment. Besides the fovea centralis, studies show that the driver's peripheral vision may also provide essential information concerning visual attention. For example, Summala et al. [81] showed that experienced drivers are able to keep the lane just by depending on their peripheral vision. Further, on an empty highway in the dark, peripheral vision is sufficient to perceive differences in light intensity indicating approaching traffic. However, since this study focuses on drives in daylight and it is assumed that none of the test subjects has experience applying peripheral vision in conditionally automated driving scenarios, the influence of peripheral vision on the driver's visual attention will be neglected.

Note that the driver's gaze behavior and visual attention are subject to individual and situational differences. For example, Konstantopoulos et al [82] compared the visual search strategies of drivers' with different levels of driving experience and showed the modification of the gaze behavior of the drivers according to visibility conditions. These results show that both external factors such as visibility conditions and internal factors such as driving experience influence the gaze behavior and, therefore, the visual attention of the driver and need to be taken into account. Typical gaze parameters used for detecting visual inattention in the driving environment are the number or frequency of off-road gazes and the mean and maximum duration of the off-road gazes [83], [84], [85]. An important step for clarifying the use of the off-road gazes was introduced by Peng et al. in [83]. An initial separation of drivers into two classes based on gaze behavior, namely low-risk and high-risk, was possible based on the maximum duration of their off-road gazes.

In conditionally automated driving scenarios, the driver is no longer responsible for the driving task and, therefore, may look away from the road the entire time. Nevertheless, the gaze behavior still contains crucial information about the visual attention of the driver, since interesting phenomena may occur when the driver performs visual secondary tasks.

In such situations, most drivers tend to gaze towards the road and the mirrors for a few seconds before continuing with the interrupted task as illustrated in Figure 4.1. These will be subsequently be referred to as *Eyes-on-Road gazes* or just *EoR gazes*. The reasons for these EoR gazes are not sufficiently investigated. One possible explanation for this behavior could be other traffic participants attracting the attention of the driver or fading attention towards the secondary task. However, it can also be assumed that drivers who are used to driving manually perform gazes towards the driving task to reorientate themselves. This assumption is further underpinned by the results of Zeeb et al. in [11]. In a large-scale driving simulator study of 107 participants, the drivers experienced multiple take-over situations while driving in a conditionally automated scenario. Zeeb et al. categorized these drivers into three groups similarly to [83] and analyzed among other things the impact of the EoR gazes on take-over performance. It could be shown that drivers who tend to perform many and longer EoR gazes are able to take-over in an appropriate way and usually in a shorter time interval than drivers who are focused solely on their secondary task. Based on these findings, parameters of the EoR gazes will be used as input for the later classification of the take-over readiness.



(a) Eyes off-road

(b) EoR gaze

Figure 4.1: The driver performs an EoR gaze by interrupting the secondary task and looking up at the road ahead before continuing with the task. In both images, two black-white markers mounted beside the steering wheel can be seen which may be used for detecting gazes in the corresponding areas.

In summary, the driver's visual perception provides the most crucial information for driving scenarios, presaging the severe consequences of visual distraction. The visual attention or distraction can be assessed over the driver's eye movement behavior and gaze direction which are subject to inter- and intra-individual differences. These findings apply for both non-automated and conditionally automated driving scenarios. As previous studies show, EoR gazes seem to be a promising indicator for visual attention, situation awareness, and maybe also for the driver's take-over readiness in conditionally automated driving scenarios.

4.2 Eyes-on-Road Detection - State-of-the-Art

Ensuring that the driver is focused on the road and not distracted from the driving task is the main goal of various systems for Eyes-on-Road (EoR) detection known from literature [86] and implementations of series vehicles [87]. All studies mentioned in Section 4.1 performed a manual EoR detection or used an offline method with special markers positioned in the scene (e.g. quadratic black-white markers beside the steering wheel as seen in Figure 4.1). Automated EoR systems usually combine an absolute gaze direction[1] with AoIs of known vehicle coordinates. If the estimated gaze hits the defined AoI, e.g. the windshield, it is concluded that the driver's visual attention is directed at this area. This method is often described as the *geometric method* since it makes use of a 3-D model of the vehicle. However, an accurate absolute gaze direction usually requires an end-of-line calibration of the camera in the vehicle, an expensive and time-consuming step. Moreover, a re-calibration becomes necessary after a certain amount of time. This effect is described, for example, in an extensive field study by Kircher et al. [88]. To avoid calibration while still applying the described approach, Vicente et al. proposed a vision-based system relying on robust facial feature tracking, head pose estimation, and a model-based gaze estimation [89]. As mentioned in Section 2.3.3, such model-based eye-tracking approaches cannot exceed an accuracy of 5° due to individual differences of the fovea centralis. Thus, this approach will most likely show limitations for small and densely packed AoIs. Note that the results shown in [89] were achieved in a stationary vehicle.

Besides the challenge of a calibration-free system, a high detection rate of EoR gazes for the classification of take-over readiness is a difficult task on account of several factors. Estimated gaze direction is inaccurate due to varying lighting conditions, face and eye structures of different ethnic groups, or optical aids. To compensate for poor gaze estimation, Tawari et al. introduced a novel framework to estimate a coarse but more robust gaze direction of the driver [86]. The authors argue that such a coarse gaze direction is sufficient for an EoR system. The authors focused on increasing robustness of the EoR detection by incorporating head pose, eyelid, and iris features over a Support Vector Machine (SVM) to a gaze-surrogate estimation and showed a significant improvement over the detection rate using only a head pose based on the data of a field study. In literature, such approaches are

[1] Absolute gaze direction refers to a gaze direction with regard to a world or vehicle coordinate system.

described as *learning-based methods*. However, learning-based methods are limited to a maximum accuracy of 5° similar to the model-based approach above. Vasli et al. extended the approach proposed in [86] by incorporating a multi-plane geometric model of the gaze zones resulting in a hybrid EoR detection method [90].

In addition to the above mentioned challenges, the driver's eyes may not be visible to the camera system, e.g., due to large head rotations. Moreover, a first generation of driver camera systems may not include eye-tracking functionalities. These scenarios require a fallback strategy to compensate for missing gaze direction. The typical fallback strategy for EoR in case of missing eye gaze signals is to use an estimated head pose. Smith et al. [91] presented a system for determining the drowsiness level and visual attention of the driver based on eye features and a gaze direction computed by means of a mono camera. The actual classification of the visual attention level was done by means of three finite state machines. If drivers rotate their head up, down, to the left, or to the right or have their eyes closed for more than a defined number of frames, a low visual attention is classified. Trefflich [92] classified a driver as attentive if the vector describing the driver's head pose intersects with a defined AoI on the windshield for at least 0.5 ms. This region on the windshield moves dynamically inside a larger static AoI and may change its size with regard to the current traffic situation. That way the AoI may increase its size towards the corresponding side when driving in a curve. If the head direction of the driver is outside of the defined AoI for more than 1.5 s, the driver will be classified as distracted. However, the absolute head pose may also be insufficient for the EoR detection in some cases since EoR gazes are usually divided into two components: eye movement and head movement. Head movement is often small and may result in an absolute head pose not necessarily facing the actual AoI when eye gaze direction is neglected.

There are already systems on the market which try to detect the driver's visual attention. In 2009, Lexus introduced the Advanced Pre-Crash Safety System using a driver camera to detect the facial direction of the driver [87]. If a driver is not looking at the road for longer than a certain threshold, an audiovisual warning is given, followed by cautionary braking. The in-vehicle camera system contained a CCD[2] imager with six near-infrared LEDs mounted on the top of the steering wheel column. The algorithm of this system actually extracts facial features of the driver's face and estimates the corresponding center line of the face. If the driver is not facing towards the road, the system will detect an unsymmetrical face in the recorded image. In case of drivers wearing glasses, features extracted from the edges of the glasses are used instead of the features of the eyes and eyebrows. Although it is an interesting approach, the described system does not calculate an accurate head direction or even a gaze direction for the EoR detection and is therefore prone to errors and false detections. For example, the Advanced Pre-Crash Safety System tends to unnecessary warnings in curvy road sections, since drivers usually focus on the apex of a curve while driving. Although these drivers focus on the road, the system detects a facial direction which is not pointing straight ahead and, therefore, is interpreted as "Driver is

[2] charge coupled device

distracted" by the system.

4.3 EoR and Control Gaze Detection based on Clustering

Geometric EoR methods represent the typical approach to detect gazes into defined AoIs, e.g. the windshield, as explained in the previous section. However, they require a calibration of the eye tracker with regard to the vehicle coordinate system, which is an expensive and time-consuming production-step. Furthermore, some tasks may be performed on handheld devices in a conditionally automated setting, thus rendering AoIs based on fixed coordinates useless. None of the approaches known from literature apply to this use-case. In this work, dynamic clusters will be learned and used to describe the AoIs by means of a distribution of a relative gaze direction[3] of the driver, so that the disadvantages mentioned above can be avoided.

4.3.1 Approach

A cluster is represented as a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ of the gaze direction $\boldsymbol{\Phi}$ which is given by the vector containing the pitch angle θ and the yaw angle ψ of the gaze direction. The parameters of the distributions are learned in different phases of a drive while the subject is focusing on the corresponding AoI. To explain the basic approach, four exemplary AoI clusters shall be distinguished: *windshield*, *handheld*, *hands-free*, and *unknown*. The cluster *windshield*

$$p^{(w)}(\boldsymbol{\Phi}) \sim \mathcal{N}(\boldsymbol{\mu}^{(w)},\boldsymbol{\Sigma}^{(w)}) \tag{4.1}$$

comprises all gazes towards the windshield, the instrument cluster display, and exterior and interior mirrors, which are mainly performed during the manual driving phases or during automated driving scenarios without secondary task. While driving manually and focusing on the cluster *windshield*, the mean vector $\boldsymbol{\mu}^{(w)}$ and the covariance matrix $\boldsymbol{\Sigma}^{(w)}$ of this cluster will be updated for each sample i. The cluster *handheld*

$$p^{(hh)}(\boldsymbol{\Phi}) \sim \mathcal{N}(\boldsymbol{\mu}^{(hh)},\boldsymbol{\Sigma}^{(hh)}) \tag{4.2}$$

contains the gazes performed during secondary tasks running on a handheld device, whereas the gazes performed during secondary tasks on an integrated, hands-free system are assigned to the cluster *hands-free*

$$p^{(hf)}(\boldsymbol{\Phi}) \sim \mathcal{N}(\boldsymbol{\mu}^{(hf)},\boldsymbol{\Sigma}^{(hf)}). \tag{4.3}$$

The mean vectors $\boldsymbol{\mu}^{(hf)}$ and $\boldsymbol{\mu}^{(hh)}$ and the covariance matrices $\boldsymbol{\Sigma}^{(hf)}$ and $\boldsymbol{\Sigma}^{(hh)}$ of the clusters *hands-free* and *handheld* are learned during automated driving phases while gazes into the corresponding AoI are detected. For all remaining gazes which cannot be assigned to one

[3] Relative gaze direction refers to a calibration-free gaze direction with regard to a camera coordinate system.

of the two clusters with an acceptable probability, the cluster *unknown*

$$p^{(u)}(\mathbf{\Phi}) \sim \mathcal{N}(\boldsymbol{\mu}^{(u)}, \mathbf{\Sigma}^{(u)}). \tag{4.4}$$

is generated. The parameters $\boldsymbol{\mu}^{(u)}$ and $\mathbf{\Sigma}^{(u)}$ of the last cluster *unknown* are updated for each sample regardless of the level of automation, because it represents the minimum probability for the assignment of a sample point to one of the defined clusters. These mentioned AoIs are pooled in the set

$$\chi = \{w, hh, hf, u\}. \tag{4.5}$$

For the estimation of these mean vectors and variance matrices, a sample estimator can be applied, as defined in

$$\boldsymbol{\mu}_i^{(\chi)} = \begin{cases} \boldsymbol{\mu}_{i-1}^{(\chi)} + \frac{\mathbf{\Phi}_i - \boldsymbol{\mu}_{i-1}^{(\chi)}}{\omega} & \text{if in learning phase} \\ \boldsymbol{\mu}_{i-1}^{(\chi)} & \text{otherwise} \end{cases} \tag{4.6}$$

$$\mathbf{\Sigma}_i^{(\chi)} = \begin{cases} \begin{bmatrix} \hat{\mathbf{\Sigma}}_i^{(\chi)}(\theta_i, \theta_i) & \hat{\mathbf{\Sigma}}_i^{(\chi)}(\theta_i, \psi_i) \\ \hat{\mathbf{\Sigma}}_i^{(\chi)}(\psi_i, \theta_i) & \hat{\mathbf{\Sigma}}_i^{(\chi)}(\psi_i, \psi_i) \end{bmatrix} & \text{if in learning phase} \\ \mathbf{\Sigma}_{i-1}^{(\chi)} & \text{otherwise} \end{cases} \tag{4.7}$$

with

$$\begin{aligned} \hat{\mathbf{\Sigma}}_i^{(\chi)}(X_i, Y_i) =& (1-\omega) \cdot \hat{\mathbf{\Sigma}}_{i-1}^{(\chi)}(X_{i-1}, Y_{i-1}) + \dots \\ & \dots \omega \cdot (X_i - \mu_{i,X}^{(\chi)}) \cdot (Y_i - \mu_{i,Y}^{(\chi)}) \qquad X, Y \in \{\theta, \psi\}. \end{aligned} \tag{4.8}$$

The choice of the weighting factor ω in (4.6) and in (4.8) is crucial for the performance of the detection of the AoI. In case of a low weighting, even short glances away from the actual AoI, e.g. when performing control gazes towards the street for re-orientation, will instantly lead to large values in the covariance matrix and shifts of the mean value of the distribution of the tablet cluster. On the other hand, large weighting factors will reduce the adaptability of the distributions to long-term changes of the position of the handheld device. A weighting factor $\omega = 10$ proved to be an effective choice. Note that the choice of ω depends significantly on the corresponding data set. Furthermore, the sample estimators need appropriate initial values to converge quickly. These initial values were calculated for each subject separately based on the data of the remaining subjects. The updated parameters are reset to the initial values for each interruption due to an occurring take-over situation for the cluster *hands-free*, since an altered position of a handheld device can be assumed after each interruption. The current sample of the gaze direction is assigned to the cluster with the highest probability

$$p^{max}(\mathbf{\Phi}_i) = max\big(p^w(\mathbf{\Phi}_i), p^{hh}(\mathbf{\Phi}_i), p^{hf}(\mathbf{\Phi}_i), p^u(\mathbf{\Phi}_i)\big) \tag{4.9}$$

$$AoI(\mathbf{\Phi}_i) = \begin{cases} 1 & \text{if } p_{max}(\mathbf{\Phi}_i) = p^{w}(\mathbf{\Phi}_i) \\ 2 & \text{if } p_{max}(\mathbf{\Phi}_i) = p^{hh}(\mathbf{\Phi}_i) \\ 3 & \text{if } p_{max}(\mathbf{\Phi}_i) = p^{hf}(\mathbf{\Phi}_i) \\ 4 & \text{else} \end{cases}. \tag{4.10}$$

The signal $AoI(\mathbf{\Phi}_i)$ contains high frequency parts representing the EoR gazes of the driver. To extract these gazes, a lowpass-filter is applied to the clusters $p^{(w)}, p^{(hh)}, p^{(hf)}$, and $p^{(u)}$ which are then used to calculate AoI_{low} by means of (4.9) and (4.10). The generated signal AoI_{low} no longer contains any EoR gazes. Instead, these gazes can be explicitly extracted through the following equation

$$EoR(\Phi_i) = \begin{cases} 1 & \text{if } AoI(\Phi_i) = 1 \;\&\; AoI_{low}(\Phi_i) = 2 \\ 1 & \text{if } AoI(\Phi_i) = 1 \;\&\; AoI_{low}(\Phi_i) = 3 \\ 0 & \text{else} \end{cases} \tag{4.11}$$

where an EoR gaze is detected whenever the lowpass-filtered signal AoI_{low} suggests that the driver is focusing on a handheld $(AoI_{low} = 2)$ or hand-free device $(AoI_{low} = 2)$ while the raw signal AoI indicates a gaze on the road.

4.3.2 Evaluation

To evaluate the proposed clustering approach, the performed EoR gazes of each experiment of the KoHAF study were video-labelled by two raters. In detail, the start and ending points of all EoR gazes in the one minute-long interval before each take-over situation was labelled. The start and ending point of an EoR gaze was defined as the first and last sample of either the head or the eye movement of the performed gaze shift. In total, the 81 test subjects performed 1199 EoR gazes. In Figure 4.2, the 2×2 confusion matrix with the classes *EoR gaze* and *no EoR gaze* is shown. The class *no EoR gaze* refers to the intervals between two sequential EoR gazes where no gazes at the road are performed. As can be seen, 445 EoR gazes have been correctly classified while there are 312 false detections and 754 missed gazes. This results in a low recall and precision value of 37% and 59%, respectively. However, the reason for this poor performance with an accuracy of only 55.5% is not due to an erroneous behavior of the proposed clustering approach. As described in Subsection 2.5.5, only a few data samples were available for larger head rotations after the calibration step preventing a reasonable estimation of function f for large gaze shifts, e.g., EoR gazes. Hence, a further evaluation of the clustering-based EoR approach based on this data is meaningless. Instead, the approach will be evaluated in Subsection 4.5.2 based on the data of the real-driving study. At this point, the need for a fallback strategy to detect EoR gazes for the classification of the take-over readiness based on the KoHAF study becomes obvious.

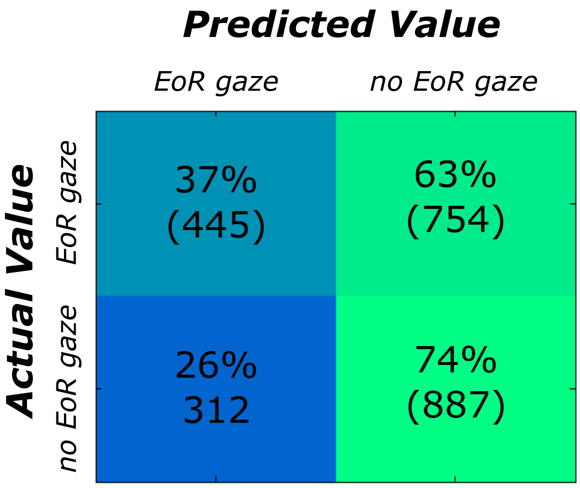


Figure 4.2: Confusion matrix of the EoR detection based on the clustering approach. The actual EoR gazes were video-labelled for the 60 s interval before each take-over situation of the KoHAF study.

4.4 Fallback Strategy for Detecting EoR Gazes

As shown in the previous subsection, the estimated gaze direction of most subjects of the KoHAF study is useless for the detection of EoR gazes. Although this was not intended, this result shows the susceptibility of the estimated gaze for applications in the vehicle and further highlights the necessity for a fallback strategy to detect EoR gazes in case of a missing or deteriorated gaze direction. The subsequently proposed fallback strategy follows the author's published patent [13] and the detection of intended head movements described in the paper [14].

4.4.1 Approach

The fallback strategy is derived from a method for detecting steering events [93] and makes use of the proposed eye movement classification MERCY introduced in Chapter 3. In Figure 4.3, a typical signal shape of an EoR gaze can be seen. The corresponding head rotation of the EoR gaze usually forms a peak with three phases: start of the head movement, stationary or static phase, and end of the head movement. This shape and movement behavior resembles recorded eye blinks as reported by Ebrahim et al. [94]. The detection is performed on the Euclidean distance of the pitch and yaw angle of the head rotation

$$\varphi_{eucl} = ||\psi_{veh} + \theta_{veh}||. \tag{4.12}$$

The head movement velocity is given by the derivation of φ_{eucl} calculated by means of the Savitzky[4]-Golay[5] filter

$$g_i = \sum_{n=-n_L}^{n_R} c_n h_{i+n} \qquad n_R = n_L = \left\lfloor \frac{m+1}{2} \right\rfloor \tag{4.13}$$

with the coefficients c_n for the first derivative, $m = 11$ sampling points, and h as first degree polynomial for the i-th data point. The Savitzky-Golay filter can be seen as a digital convolution filter which is continuously fitting polynomials of a specified degree to adjacent samples of a given signal. Moreover, the filter can be used to estimate the differentiation of the input signal if the correct coefficients are used [95].

In an initial step, the proposed approach detects movements of the head based on a specified minimum angle $\varepsilon_{amp,init}$ and velocity $\varepsilon_{vel,init}$ as shown in the upper plot of Figure 4.3. Note that these minimum thresholds are empirical values and must be selected depending on the accuracy of the applied measurement system and with respect to the quality of the recorded data. For the setup used in this work, these values were set to 4° and $6°/s$. The first sample at which the velocity exceeds the threshold $\varepsilon_{vel,init}$ will define the starting point of the EoR gaze. The starting phase ends as soon as the velocity drops below $\varepsilon_{vel,init}$ again. At this point, the static phase starts and it is assumed that the driver is focusing on the road. The EoR gaze is completed by a head movement back to the original position, i.e. to the secondary task, defined as the ending phase. This phase is given by the interval during which the velocity is below the threshold $-\varepsilon_{vel,init}$. This step provides a first estimation of which signal events correspond to intended, actual head movements rather than just being artifacts or noise.

In the subsequent step, the detected events are made plausible by means of minimum angle $\varepsilon_{amp,75}$, minimum velocity $\varepsilon_{vel,75}$, and minimum duration thresholds ε_{rest} and ε_{static} as well as by the rotational direction. The thresholds for the head angle and velocity are adapted online over the corresponding 75%-quantile calculated over all detected events of the previous step as suggested by Galley et al. [93]. The calculated head amplitude and velocity have both to exceed these thresholds or the event is discarded. The minimum duration of the static phase during the EoR gazes was set to $\varepsilon_{static} = 200\,\text{ms}$ according to an analysis of recorded EoR gazes. Similarly, the idle period after each detection ε_{rest} is analyzed and events occurring in this specified interval are ignored. The algorithm assumes a resting phase of at least 200 ms. Events below these duration thresholds ε_{static} and ε_{rest} usually refer to head gestures such as shaking, e.g., when negating a question. The last plausibility check involves the direction of the head rotation. Only movements upwards, to the left, or combinations of those are considered as possible EoR gazes. This step is performed on the original yaw ψ_{veh} and pitch angle θ_{veh}. This is an additional

[4] Abraham Savitzky, * 29. May 1919 in New York City, USA, † 05. February 1999 in Naples, Florida, was an american analytical chemist.

[5] Marcel Jules Edouard Golay, * 03. Mai 1902 in Neuchâtel, Switzerland, † 27. April 1989 in Lausanne, swiss mathematician, physicist, and information theorist.

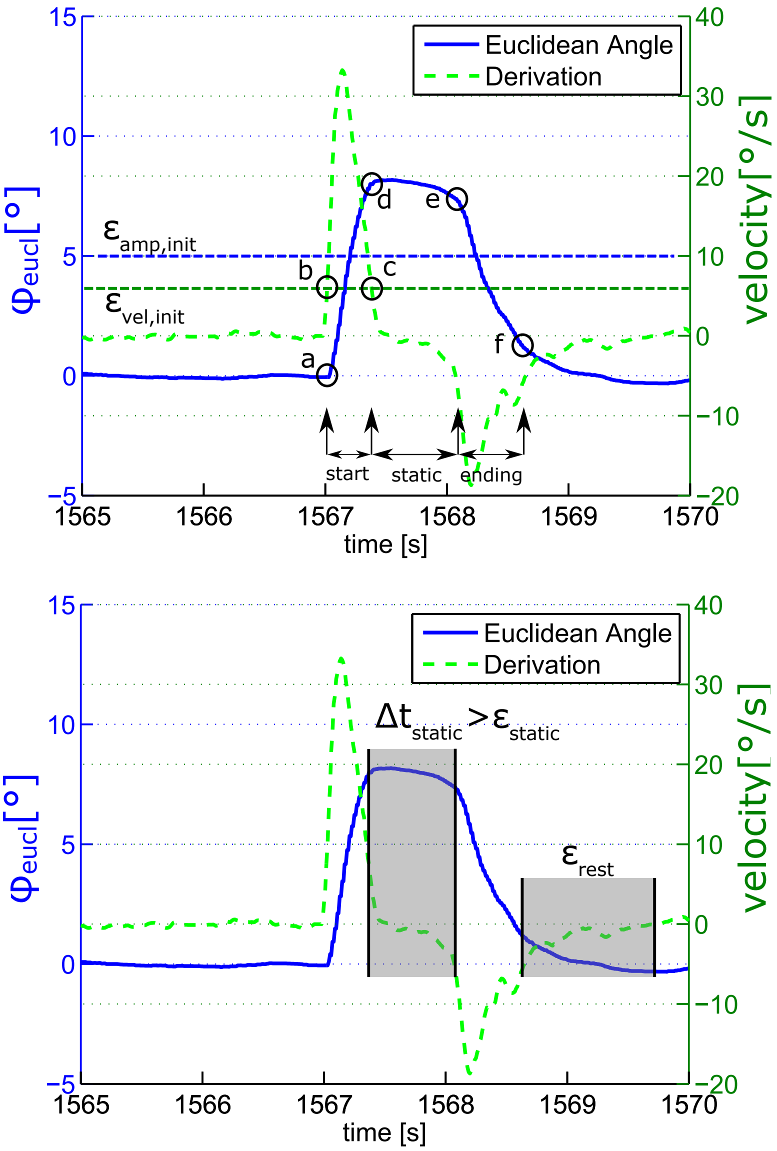


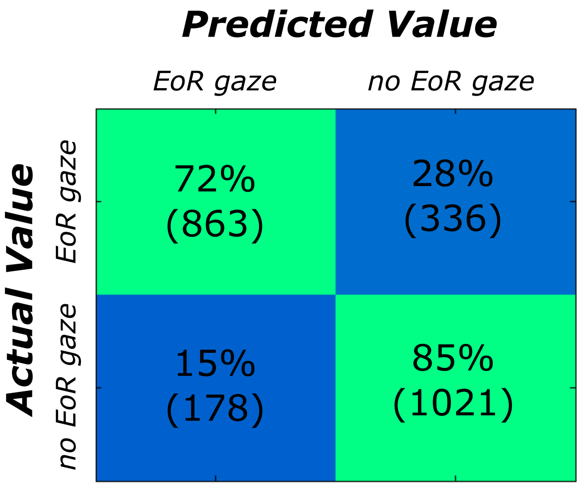
Figure 4.3: Visualization of the necessary steps of the fallback strategy. In the upper plot, the typical phases of the head rotation during an EoR gaze are shown, including the initial thresholds used for detecting possible EoR gazes. The start- and endpoints of each phase are defined by the velocity threshold $\varepsilon_{vel,init}$. For example, at point b) the velocity exceeds the threshold which corresponds to the beginning of the start phase of the possible EoR gaze marked with a). Similar, at point c) the velocity falls below the threshold corresponding to the end of the start phase marked with d). The same procedure is repeated with the negative thresholds for the beginning e) and endpoint f) of the ending phase. In the lower plot, the idle periods during and after each possible EoR gaze used as a condition in the plausibility check are highlighted.

condition for further reducing false detections since movements in the opposite direction usually do not correspond to EoR gazes.

The described approach provides a fallback strategy in case of no gaze direction is available. However, if some form of eye-tracking is also possible, independent of the available quality, MERCY is applied to classify occurring saccades online. Similar to the approach described above, the detected saccades are made plausible using an amplitude threshold adapted online over the 80%-quantile of all classified saccades and the direction of the saccade. Saccades with a large amplitude in the direction of the windshield, i.e. upwards and/or to the left are considered as possible EoR gaze. However, this additional analysis of the eye movements is only performed if the eye tracker assures the validity of the current data sample. The head-based approach may influence the detection based on the eye-tracking data by reducing the amplitude threshold from 80% to the 75%-quantile if a head movement which detected simultaneously is discarded due to a low amplitude.
Note that this fallback strategy can only be performed with appropriate accuracy if there is some information given concerning the automation level or regarding the driver's current focus. For example, if the driver is currently focusing on a display located at the center console this head pose can be used as an initial state for the approach. That means that in this situation the fallback strategy may begin to analyze the head movements according to the above described approach. Without such information to determine a starting point, a gaze into the left exterior mirror may be recognized as EoR gaze although the driver is focusing on the road all the time. Such information could be provided by the vehicle's CAN signals describing keystrokes on the multimedia unit which typically correlate to fixations on the corresponding buttons.

4.4.2 Evaluation

Similar to the evaluation of the clustering approach in Subsection 4.3.2, which performed quite poorly when detecting the EoR gazes of the KoHAF study, the proposed fallback strategy was completely implemented in Simulink and applied to the same data. Again, the detected EoR gazes were compared to the video-labelled EoR gazes for the 60 s intervals before each take-over situation while the class *no EoR gaze* represents the intervals between two sequential EoR gazes, i.e. the gaze-free periods.
In Figure 4.4, the corresponding confusion matrix of the fallback strategy based only on the head movements can be seen. In total, 863 of the 1199 EoR gazes have been detected resulting in a recall value of 72%. Hence, the number of correctly detected EoR gazes could be nearly doubled compared to the 445 EoR gazes correctly classified by means of the clustering approach. At the same time, the false detections decreased from 312 to 178 falsely detected EoR gazes, resulting in a precision value of 82.9%. Moreover, the fallback strategy shows an accuracy of 78.5% which corresponds to an increase of 23% compared to the clustering approach. However, it is necessary to mention that the number of true positives cannot be increased much further since most of the remaining 336 EoR gazes were performed without any detectable head movements. Hence, these EoR gazes may only be recognized using eye-tracking data.



Actual Value \ Predicted Value	EoR gaze	no EoR gaze
EoR gaze	72% (863)	28% (336)
no EoR gaze	15% (178)	85% (1021)

Figure 4.4: Confusion matrix of the EoR detection based on the fallback strategy considering only head movements. The actual EoR gazes were video-labelled for the 60 s interval before each take-over situation of the KoHAF study.

This is why the following evaluation applied the second variant of the fallback strategy based on the detected head and eye movements. As shown in Figure 4.5, considering large saccades classified by MERCY enables an even higher recall of 83% compared to the approach solely based on head movements. However, the analyzed eye-tracking data is of low quality and therefore involves the risk of additional false detections. This is the reason why the precision value decreases to 76% and the number of false detections again rises to 26%, equal to the original value of the applied clustering approach. The overall accuracy of the fallback strategy considering the eye movements equals 78.4%.

In summary, both variants of the fallback strategy perform significantly better than the clustering approach, which suffers from the low quality of the estimated gaze. Hence, the benefit of a fallback strategy to detect EoR gazes in case of a missing or deteriorated gaze direction could be highlighted. There is no difference in the overall performance of the two variants with regard to the accuracy values. However, the variant which considers only the head rotation is obviously limited by the number of EoR gazes which contain a detectable movement of the head. That means that further significant improvements of the detection rate of the EoR gazes might be unlikely. Hence, a recall of about 80% can be seen as general limitation of EoR detection systems based solely on head movements. At least rudimentary eye-tracking data is necessary to enable a further improvement of the recall as shown with the second variant of the fallback strategy. However, this recall improvement comes with a decrease in precision and an increase of the number of false detections due to the deteriorated quality of the estimated gaze.

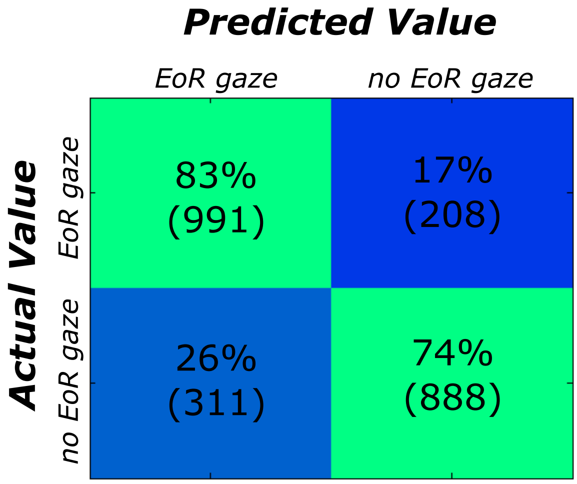


Figure 4.5: Confusion matrix of the EoR detection based on the fallback strategy considering head and eye movements. The actual EoR gazes were video-labelled for the 60 s interval before each take-over situation of the KoHAF study.

4.5 Detecting Eyes-on-Road online in the Vehicle

As mentioned above, the fallback strategy was completely implemented in Simulink and could be applied directly online in the vehicle over the dSpace ControlDesk. On the other hand, the EoR detection based on clustering was initially developed in an offline MATLAB environment. Hence, some parts of the approach mentioned in Section 4.3 need to be slightly reconsidered for an online approach in the vehicle. Moreover, this approach could not be evaluated fairly based on the data of the KoHAF study due to the poor estimated gaze. Therefore, the classification performance of this approach is evaluated on the data of the real-driving study described in Subsection 2.4.5. In this study, a near-to-production driver camera was applied to estimate the driver's gaze resulting in data quality which can be expected in later series vehicles.

4.5.1 Modifications

For the offline evaluation in Subsection 4.3.2, it was only necessary to introduce four clusters representing the four AoIs *windshield*, *handheld*, *hands-free*, and *unknown*. However, one aspect to investigate was how well the clustering approach is able to handle multiple as well as smaller AoIs which are typical for the vehicle environment. As a consequence, the original AoI *windshield* was separated into the AoIs *windshield*, *left mirror*, *right mirror*, and *interior mirror*. Further, the AoI *unknown* was replaced by a fixed, scalar threshold. This has a practical reason since the cluster *unknown*, in particular the variance of this cluster, tend to spread in case of the real-driving study as well as for pre-studies over a large area. As a result, the actual probability for this cluster decreased so significantly that it was usually never classified even in areas between AoIs. Figure 4.6 shows all of the analyzed clusters visualized in a vehicle interior and summarized by the set

$$\chi \in \{w, hh, hf, im, lm, rm\}. \tag{4.14}$$

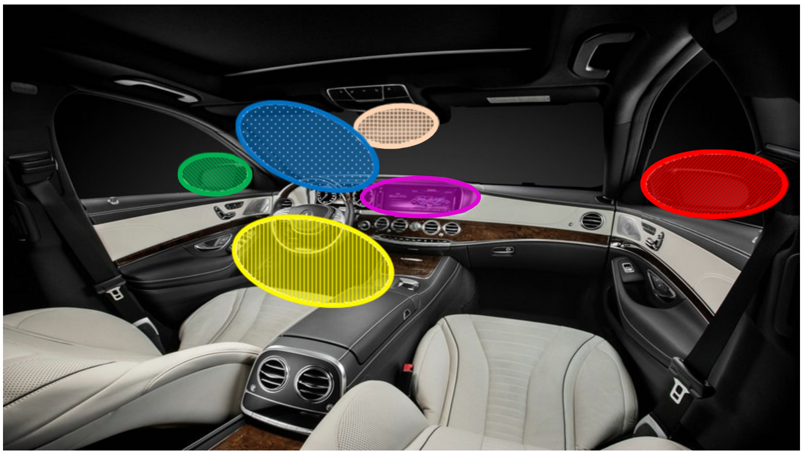


Figure 4.6: Exemplary visualization of all analyzed clusters in the vehicle interior. The clusters are: windshield (blue, dotted area), interior mirror (beige, checkered area), left mirror (green, diagonally dashed area), right mirror (red, diagonally dashed), hands-free device (purple area without a pattern), handheld device (yellow, vertical dashed area).

Moreover, the offline evaluation calculated the initial values $\boldsymbol{\mu}_0^{(\chi)}$ and $\boldsymbol{\Sigma}_0^{(\chi)}$ for the mean and covariance of the clusters over the MATLAB class *gmdistribution* for each subject individually. In more detail, the KoHAF study can be separated into four route sections, namely: the introductory section, the conditionally automated section between the introductory section and the first take-over situation, and the conditionally automated sections between the first and second as well as between the second and third take-over situation. For these four sections, four different pairs of initial values were used for each cluster by calculating the mean and the covariance over the complete corresponding route section. Obviously, this is an inadequate approach for online scenarios. Hence, for the online approach a priori learned initial parameters derived from three subjects[6] for each cluster were used. The subjects were recorded while they were looking into the different clusters for several seconds. Based on these data sets, the initial mean and covariance values were estimated and used for the online approach. For applications in series vehicles, this number of used subjects

[6] These subjects did not participate in the real-driving study.

to generate the initial values should be increased further to obtain a preferably widespread driver pool in terms of body height, seating position, and gaze behavior.

An additional topic when considering the adaption of the cluster-based approach to online scenarios is the overlapping of clusters. Since the clusters are updated continuously after the initialization and there is no resetting of the mean and covariance values during a drive, the chance of overlapping clusters increases significantly compared to the offline evaluation. For example, if a driver would move the handheld device in front of the integrated hands-free system while driving in a conditionally automated scenario, the mean value of the handheld cluster would approximate the mean of the hands-free cluster. As a consequence, all subsequent gazes would probably be assigned to the handheld cluster even if the driver interacts with the integrated device. To prevent overlapping, the online variant integrates boundaries for each cluster except for the cluster *handheld*. These boundaries limit the range of motion of the mean values of the corresponding cluster. The upper and lower boundaries are defined with respect to the initial mean values and a tolerance value $\boldsymbol{\xi}$ by

$$\boldsymbol{b}_{+}^{(\chi)} = \boldsymbol{\mu}_{0}^{(\chi)} + \boldsymbol{\xi} \quad \text{und} \quad \boldsymbol{b}_{-}^{(\chi)} = \boldsymbol{\mu}_{0}^{(\chi)} - \boldsymbol{\xi}, \qquad \boldsymbol{\xi} = \begin{pmatrix} 10^\circ \\ 10^\circ \end{pmatrix} \in \mathbb{R}^2. \tag{4.15}$$

By means of these boundaries $\boldsymbol{b}_{+}^{(\chi)}$ and $\boldsymbol{b}_{-}^{(\chi)}$ the mean and covariance is limited during the updating step over

$$\boldsymbol{\mu}_{n}^{(\chi)} = \begin{cases} \hat{\boldsymbol{\mu}}_{n}^{(\chi)} & \text{if } \hat{\boldsymbol{\mu}}_{n}^{(\chi)} < \boldsymbol{b}_{+}^{(\chi)} \quad \text{and} \quad \boldsymbol{b}_{-}^{(\chi)} < \hat{\boldsymbol{\mu}}_{n}^{(\chi)} \\ \hat{\boldsymbol{\mu}}_{n-1}^{(\chi)} & \text{else} \end{cases} \tag{4.16}$$

where $\hat{\boldsymbol{\mu}}_{n}^{(\chi)}$ describes the estimated value of the mean of this current iteration n. This tolerance value seemed to be an appropriate choice based on earlier experiments. For future work, instead of determining one common tolerance value for all clusters experimentally there should be an individual tolerance value for each cluster considering the extreme cases of the widespread driver pool used to learn the initial values. The cluster *handheld* is not limited in its range of motion but is considered with lowest priority. That means that in case of the overlapping scenario described above, the cluster *handheld* might have an overlapping area with the cluster *hands-free* but it would be never classified as currently focused cluster due to the lower priority. Thus, the other AoIs represent a software solution of the boundaries of the cluster *handheld* without restricting its range of motion.

As a final modification of the EoR approach based on clustering, a priori probabilities were applied as mixture factors of the mixture distribution of the clusters. Similar to the mixture model described in equation (3.1) and (3.2), mixture factors π^{χ} were defined for each cluster representing a weighting factor for each distribution based on latest obtained information. These mixture factors must fulfill the condition

$$\sum_{\chi} \pi^{(\chi)} = 1 \tag{4.17}$$

at all times. Following this condition, an uniform distribution was chosen for $\pi_0^{(\chi)}$

$$\pi_0^{(\chi)} = \frac{1}{|\chi|} \in]0,1] \tag{4.18}$$

as initial values weighting each cluster equally. As explained in Subsection 2.2.2, a fixation represents a static eye movement focusing the fovea centralis on a temporal target, e.g., one of the introduced clusters. Further as shown in Section 3.3, about 90% of eye-tracking data recorded during conditionally automated driving scenarios consists of such fixation points and only 10% of saccades. As a consequence, it can be assumed that the eyes remain focused on the same visual target for both the current sample point and for the next sample. Hence, the a priori values were increased for the time the driver is focusing on the corresponding AoI by means of a sample estimator

$$\pi_{n+1}^{(\chi)} = \omega \frac{n^2 \cdot \pi_n^{(\chi)} + n}{n^2 + n}, \quad \omega = \frac{1}{n} \tag{4.19}$$

In accordance with condition (4.17), the a priori values of the remaining clusters need to be decreased proportionately for

$$\varepsilon_n = \pi_{n+1}^{(\chi)} - \pi_n^{(\chi)} \tag{4.20}$$

which equals the increase of the a priori of the currently focused AoI between the latest sample points. To prevent the violation of the boundaries of the remaining a priori values given by $]0,1]$, this difference ε_n has to be separated among the remaining AoIs according to the proportion of their a priori values in equation (4.17). The corresponding difference is given by

$$\Delta_n^{(\kappa)} = \frac{\pi_n^{(\kappa)} \cdot \varepsilon_n}{1 - \pi_n^{(\chi)}} \tag{4.21}$$

with

$$\kappa \in \{w, hh, hf, im, lm, rm\} \quad \textit{and} \quad \kappa \neq \chi. \tag{4.22}$$

However, this approach may result in impractically low a priori values for sequences during which the driver keeps focusing the same AoI. To prevent the decrease in the a priori values of the currently not-fixated AoIs from being to extreme, each a priori value is scaled to the range $]1,2]$ by adding $+1$. Hence, the a priori value represents an amplification with a minimum greater than one. In Figure 4.7, the behavior of the a priori values is shown for gazes according to the sequence of AoIs given by

$$w \to lm \to w \to rm \to w \to im \to w \to hf \to w \to hh \to w \tag{4.23}$$

where each AoI was focused for about three seconds. In the upper plot, the sequence described in equation (4.23) can be seen clearly by reference to the probability density functions of the various clusters. In the synchronous plot below, the a priori signals are visualized. The a priori value of the windshield cluster shows a sawtooth-shaped signal since it was fixated after each cluster. Moreover, the exponential approximation of each value to the minimum and maximum can be seen.

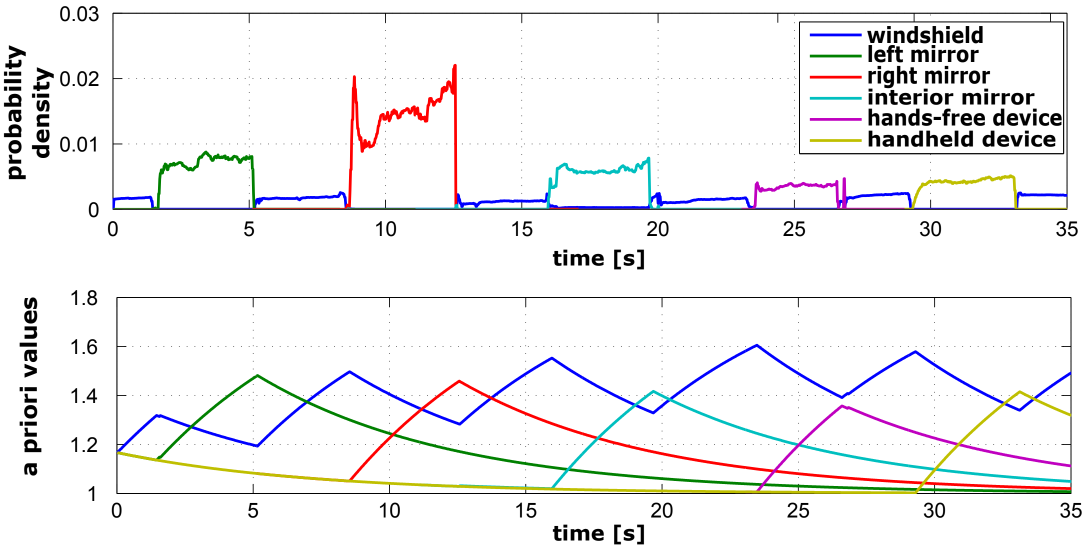
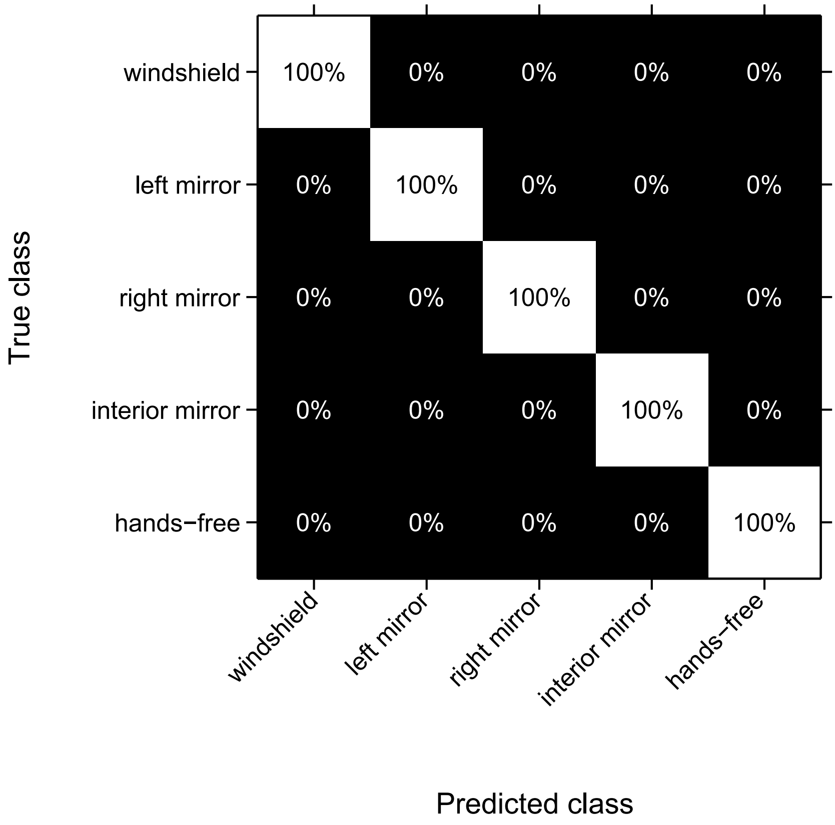


Figure 4.7: Both plots are generated for the gaze sequence given in equation (4.23). The upper plot shows the probability density of the different clusters while the lower plot visualizes the behavior of the a priori values. The signals are colored according to their corresponding cluster.

4.5.2 Evaluation of the online Approach

After adapting the clustering-based approach according to the modifications summarized in the previous Subsection, an evaluation is performed based on the data recorded during the real-driving study described in Subsection 2.4.5. In a first step, the detection accuracy of the approach with regard to the introduced AoIs will be evaluated. For this purpose, the instructed gazes at the different AoIs performed by the driver in the westbound route section of the third lap were used. In this lap, each driver was instructed to look in the AoIs *windshield*, *left mirror*, *right mirror*, *interior mirror*, and *hands-free device* for about five seconds. Note that there was always a short break between two sequential instructions during which the driver usually focused on the windshield. These intervals between two sequential AoIs were not used for the evaluation. For ground truth, one of the instructors sitting on the backseat of the vehicle labelled the gazes online during the experiment by means of manual triggers. The first and last two seconds of each fixation at an AoI were discarded to guarantee synchronicity between the actual gaze start and ending and the manually set triggers.

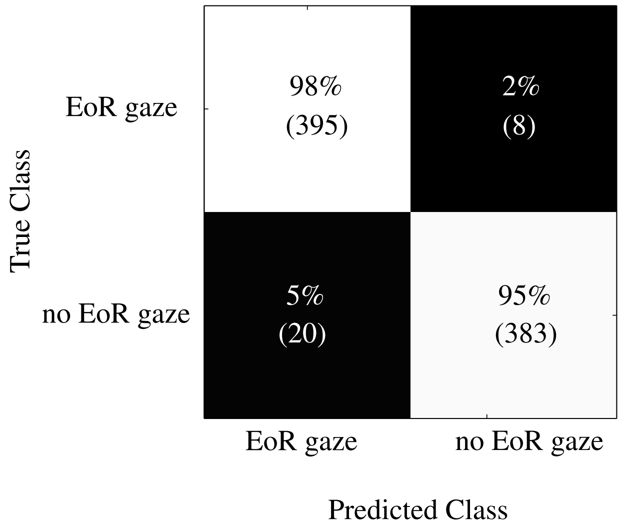
The confusion matrix in Figure 4.8 visualizes the final results with regard to the approach introduced in Subsection 4.3.1 and modified for applications in vehicles in Subsection 4.5.1. As can be seen, the approach reaches perfect accuracy. Each instructed gaze of the nine drivers was correctly assigned to the corresponding AoI for the complete fixation interval. This result highlights different aspects of the approach and the experimental setting. Obviously, the approach is able to detect distinct gazes in each of the discussed AoI robustly. Note that these gazes were performed about eight minutes after the beginning of the first lap which also represents the interval during which the clustering-based approach adapts the clusters to the individual driver. Moreover, the results show that the accuracy and robustness of the applied near-to-production driver camera is sufficient for the detection of various AoIs during conditionally automated drives during realistic lighting conditions. In addition, even the gazes in the right mirror were detected correctly. This is a remarkable result for the implemented camera system as well as for the clustering-based approach since the horizontal gaze angle to this AoI is large and requires significant head rotations.



	windshield	left mirror	right mirror	interior mirror	hands−free
windshield	100%	0%	0%	0%	0%
left mirror	0%	100%	0%	0%	0%
right mirror	0%	0%	100%	0%	0%
interior mirror	0%	0%	0%	100%	0%
hands−free	0%	0%	0%	0%	100%

Figure 4.8: Confusion matrix of the cluster detection based on the instructed driver gazes of the real driving study.

In a second step, the detection of the drivers' EoR gazes were evaluated. This evaluation was based on the EoR gazes performed by all drivers during each eastbound route section. During these sections, the driver performed secondary tasks which were interrupted by uninstructed EoR gazes. Although the used data set was not extensive, many different types of gaze behaviors could be observed. As expected, the number of performed EoR gazes varied significantly between the different subjects. In total, 403 EoR gazes were performed for a cumulated driving time of about 62 minutes. While some drivers performed up to 100 gazes during the seven minutes of conditionally automated driving time with secondary tasks, there was one driver who interrupted the performed secondary tasks for only four gazes. Moreover, drivers could be observed performing short or long EoR gazes using only their eyes or a head- and eye-coordination for gaze shifts. In addition, different head positions during the performance of the secondary tasks could be observed influencing calculated gaze direction. As Ground Truth, the EoR gazes were labelled online by one of the instructors inside the vehicle and verified offline afterwards by means of a recorded video stream.



True Class \ Predicted Class	EoR gaze	no EoR gaze
EoR gaze	98% (395)	2% (8)
no EoR gaze	5% (20)	95% (383)

Figure 4.9: Percentage and absolute number of correctly and falsely detected EoR gazes by means of the clustering approach.

The confusion matrix of Figure 4.9 shows the performance of the EoR detection based on all gazes independent of the drivers. In total, 98% of the actual 403 EoR gazes could be detected correctly. Further, the detection approach shows a false positive rate of 5%, which equals 20 falsely detected gazes. This result highlights the feasibility of robust EoR detection based on near-to-production camera systems in realistic driving scenarios. Besides a few artifacts and incorrect estimations of the gaze directions, the false positive rate of 5% can be accounted for by the defined assumption of the a priori values. As mentioned in Subsection 4.5.1, the application of the a priori values defined by (4.19) is based on the assumption that the probability is increasing for the AoI which was fixated for the previous sample. This assumption is valid and reasonable for the overall detection performance as described in the previous subsection and optimizes the detection of AoIs. However, a priori values may have a negative impact during gaze shifts since they increase the weighting of the previous AoI compared to the destination of the gaze shift. As a consequence, if the driver does not exactly focus at the center of the cluster of the subsequent AoI, such estimated gaze points could be assigned to the previous cluster with the increased weighting. This is especially critical for densely packed or large AoIs. Moreover, recorded gaze points during the actual gaze shifts will be assigned to the previous cluster at the beginning of the gaze shift although they should be assigned to none defined AoI, representing the former cluster *unknown*.

For the last step of the evaluation, the detection algorithm is applied to separate secondary tasks performed on a handheld or hands-free device. The confusion matrix in Figure 4.10 again shows an outstanding detection performance. While all samples belonging to the hands-free AoI are classified correctly, 10% of the samples recorded during the usage of the handheld device were assigned to the false class. The reason for this result is the low priority of the handheld cluster during overlapping scenarios. One of the subjects was holding the handheld device in front of the hands-free cluster which caused the assignment of the data samples to the overlapped hands-free cluster instead of the correct handheld cluster. If this subject is excluded from the evaluation, a perfect confusion matrix with 100% accuracy is reached. Obviously, overlapping scenarios are an open topic concerning the detection of AoIs in the vehicle, in particular for dynamic cluster such as the handheld cluster. Although first studies are found in literature with regard to the topic of detecting the 3D gaze position [96], the robust and accurate detection of the vergence of the driver's eyes over near-to-production camera systems in the vehicle will remain a challenging task during the next decade. Nevertheless, the overall accuracy of the AoI and EoR detection is outstanding and close to the optimal result. However, the presented driving study has limitations in terms of the number of participated subjects and their variations regarding ethnicity, gender, and optical aids since only trained drivers were allowed to participate in this study.

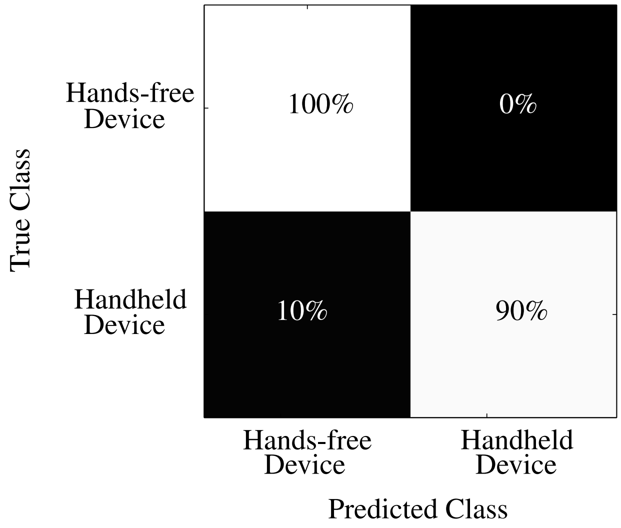


Figure 4.10: Classification performance of the clustering-based approach for separating the usage of handheld and hands-free devices based on the real driving study.

4.6 Summary

In this chapter, methods for detecting eyes on road were investigated in detail. At first, the concept of EoR gazes was introduced, it is a common phenomenon in conditionally automated driving scenarios. Following that, it was shown that there is already an extensive amount of literature and approaches available. However, none of them fulfills all the requirements for a calibration-free system for series vehicles able to detect driver gazes in dynamic AoI. Moreover, a robust fallback strategy in case of missing or invalid gaze data is required. As a consequence, the chapter proposes a novel approach for detecting eyes on road based on Gaussian mixture models. This model generates clusters representing the different AoIs such as the exterior mirrors or handheld devices. These clusters can be individually adapted to the different drivers and situations. Further, the chapter shows the approach modifications necessary to transfer it to the online scenario. Using the data of a real driving study with a near-to-production camera system, the outstanding performance of the approach was verified. Of course this approach might fail in the event of a distorted gaze estimation as shown in Subsection 4.3.2. As a consequence, a novel fallback strategy was proposed especially designed for detecting EoR gazes in conditionally automated driving scenarios. This fallback strategy analyzes the head movement behavior instead of focusing on an absolute head pose. Based on the data of an extensive driving simulator study, the benefit of this fallback strategy was presented. Nevertheless, this approach has a natural limitation since only about 80% of all EoR gazes include head movements. The proposed approaches will be used to extract features for a classification of the driver's take-over readiness in Chapter 6.

5 Automated Driver-Activity Recognition

The most significant difference between non automated and conditionally automated driving is the transfer of responsibility. While the driver is currently responsible for the vehicle at all times, an automated driving function takes over this responsibility in a conditionally automated setting. As a consequence, this new level of automation involves great benefits in terms of comfort and enables the driver to use the driving time more efficiently by allowing the performance of secondary tasks. However, many studies indicate a correlation between secondary tasks and a low take-over quality [8], [9], [97]. Hence, the classification of the take-over readiness proposed in this study considers the driver's activity as one source of information for determining if the driver is able to take-over appropriately. In order to extract relevant features of secondary tasks for this classification, automated and online-capable methods for driver-activity recognition (DAR) during conditionally automated drives need to be developed and evaluated. In the following chapter, the state-of-the-art for driver activity recognition is summarized before two different novel approaches are introduced in Section 5.2 and Section 5.3. These approaches will be evaluated and compared to each other based on the data of the KoHAF experiment in Section 5.4. Finally, in Section 5.5, the superior approach will be implemented and tested in the vehicle environment based on the data from the real driving study.

5.1 Existing Methods for Driver-Activity Recognition

Detecting the driver's secondary task is one of the various applications of human activity recognition (HAR). Other applications of HAR involve smart security surveillance [98], health monitoring [99], and efficient human-machine interfaces [100]. Many of these approaches for HAR are based on computer vision methods, i.e. using remote camera systems. A comprehensive review of HAR based on computer vision is given by Turaga et al. in [101]. Moreover, there are some studies regarding the application of in-vehicle camera systems to recognize of different types of activities inside the vehicle. Typically, the motion patterns of different body parts, e.g., motion of the hands, legs, the torso, the head, and the posture of the body are taken into account to infer driver activity. The detected activity does not have to be a concrete specified interaction with the vehicle or its surroundings, such as shifting into another gear, but it can describe an AoI the driver is currently paying attention to. These AoIs summarize all activities performed in the specified region. In this way, Ohn-Bar et al. proposed a framework capable of distinguishing between activities performed inside of three AoIs, namely the gear region, the instrument cluster, and the wheel region. To reliably distinguish between the mentioned AoIs, the authors combined hand activity recognition [102] with the estimated gaze direction. Another more general approach for using semantic driver activity analysis that could easily be extended to multiple sensor types was introduced by Park and Trivedi in [103]. The distinguished results indicate that activities in adjacent AoIs will often be confused and, therefore, depend on features of motion patterns of various body parts to improve the classification accuracy.

While approaches based on computer vision usually still have limitations in terms of distinguishing activities inside the same or adjacent AoI and with similar motion patterns, wearable sensors are able to provide highly accurate data for basic research due to their intrusiveness. At the same time, intrusiveness represents the major drawback of wearable sensors, since it usually precludes application in series vehicles and influences the natural driving behavior. Lara and Labrador [104] provided a comprehensive overview of wearable sensors and their applicability for various topics. A large number of studies discuss activity recognition based on accelerometer data [105], [106]. In [107], Sathyanarayana et al. tried to detect distracted driving behavior by using accelerometers on the driver's legs and head. It was noted that the distracted activity of using a mobile phone can be detected reliably with the intrusive head sensor. However, this sensor type is better suited to physical activities such as jogging or climbing stairs, than to the primarily cognitive tasks which are of interest for automated driving. Because human eyes have the potential to reveal various driver states such as drowsiness [108] and inattention [109], eye movement analysis has been the focus of much research in different fields of study. Eye movements were first considered as a possible information source for activity recognition by Bulling et al. [110]. To classify different office activities, such as browsing the web or reading a text, the authors recorded eye movements by means of EOG, detected basic eye patterns, namely saccades, fixations and blinks, and extracted multiple features based on these patterns [110]. The authors reported that eye movement analysis is suitable for activity recognition. Based on these findings, Banerjee et al. [111] analyzed time, frequency, and time-frequency correlated eye signal features to recognize eight different tasks. By combining all eye features from the time, frequency, and time-frequency domains, the highest classification result of 90.39% was obtained. However, note that these results were achieved on an EOG data set recorded by means of a high sampling rate of 250 Hz and a subject-dependent training set. Besides the above work based on EOG, the increasing signal quality of eye-tracking systems facilitates the recording of eye movements. Head-mounted eye trackers enabled a new approach combining eye movement analysis and visual features due to the available eye and field camera [112]. Using the Google Glass platform as a sensor to measure eye blinks and head motion patterns, Ishimaru et al. verified in [113] that these standard sensors in combination with just four features yield the potential for human activity recognition. Besides the typical visual-based tasks, such as watching a movie and reading, the analyzed activities in [113] also contained a demanding cognitive activity, namely solving mathematical problems, and a physical activity, namely sawing. According to the authors, the cognitive task was hard to classify due to its dual character: writing the answer and looking at the assignment sheet.

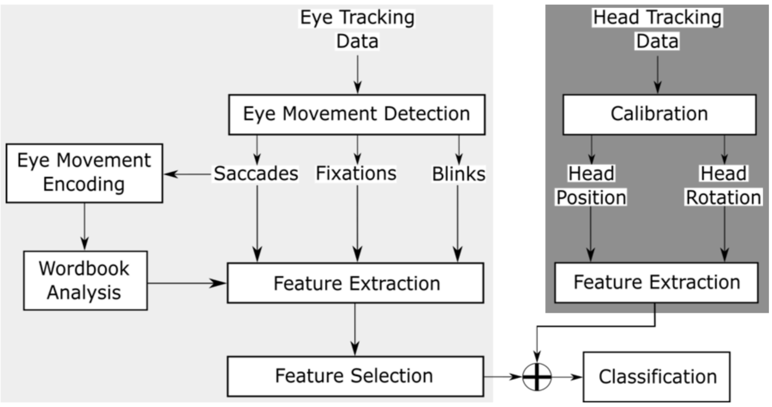


Figure 5.1: Overview of the applied architecture.

5.2 CIDAR - Chronologically Independent Features for Driver-Activity Recognition

The framework of the first approach to detect the driver's activities proposed in this work is shown in Figure 5.1. This architecture can be divided into two paths which are merged at the final classification step: the eye-tracking path for extracting features of the eye movements, which is derived from the framework introduced by Bulling et al. [110], and a novel head-tracking path for extracting features of the driver's head movements. Subsequently, the modifications of the original eye-tracking path as well as the novel head-tracking path will be described in detail. This section is based mainly on the author's publications [15] and [16].

5.2.1 Architecture of CIDAR

Starting with the eye-tracking path, shaded in light gray in Figure 5.1, head-mounted or remote camera systems are used to record the eye movements instead of an EOG system as applied in [110]. Due to the often significant lower frequency and the typical challenges concerning video-based eye trackers, e.g., changing illumination conditions, individual shape of the pupil, disturbing accessories, etc., a lower signal quality is expected. Moreover, test subjects are not forced to perform secondary tasks continuously as described in the original study. Instead, the drivers are enabled to interrupt or even change the task at will. As shown in Chapter 3, such behavior influences the eye movement behavior and has to be taken into account for the eye movement classification step. Hence, the proposed approach MERCY, described in Section 3.2, was applied to distinguish between saccades and fixations in this architecture. Due to the permanent online adaption of the two Gaussian distributions of the mixture model, the algorithm is able to adapt to the highly intra- and inter-individual eye patterns. Besides saccades and fixations, the DAR is based on detected eye blinks. In the case of a camera signal, eye blinks are usually not explicitly detected but modeled from the data [48]. The actual blink detection of this approach consists of two steps. In the first step, signal parts with a low signal quality are deleted. Therefore, a moving temporal window 5 s wide is shifted over the whole signal while calculating the amount of invalid values. Depending on the applied eye tracker, invalid samples might be recognized by means of a validity signal or specified signal values of the actual eye-tracking signal. If the percentage of invalid data is greater than 30%, this signal portion will not be considered for the next analysis step. The idea behind this is to reduce the amount of false detections in areas where the signal quality is inacceptable. In a second step, all remaining sequences in which the eye tracker was not able to detect the pupil, will be labelled as an eye blink of a certain duration t_{blink} if the following threshold criteria are fulfilled:

$$|th_{min} - tol| \leq t_{blink} \leq |th_{max} - tol|. \tag{5.1}$$

The threshold values from (5.1) were set to $th_{min} = 0.1\,\text{sec}$ and $th_{max} = 0.4\,\text{sec}$ according to [114] and represent the average minimum and maximum duration of an eye blink. The tolerance variable results from the circumstance that the duration of the examined events of invalid values typically does not correspond to the average eye blink duration. The reason for the lack of correspondence is that the eye-tracking system is still able to detect the pupil while the eyelid is already moving downwards in the closing phase or moving upwards in the opening phase and the pupil is still visible. Therefore, the tolerance value has to be chosen according to the sampling rate f of the eye tracker and is set to $tol = \frac{1}{f}$.

The combined eye movement encoding and wordbook analysis presented in Figure 5.1 perform a mapping of each saccade to a symbol depending on the amplitude and direction of the saccade. A moving window of a specified size m_{word} is shifted over the sequence of characters and all existing combinations of characters, called *words*, are detected and saved in the wordbook Wb_l as in [110]. In the next step, the features used for the classification are extracted. In total, 145 features based on the detected saccades, fixations, blinks, and wordbooks have been analyzed and will be described in more detail in Subsection 5.2.2). Due to the high number of features and therefore increased risk of an overfitting of the classifier, a feature selection is performed to reduce the number of the features. This selection step is performed by means of the *Fast Correlation-Based Filter* (FCBF) algorithm introduced by Yu and Liu in [115]. FCBF chooses a subset of features according to the redundancy and relevance analysis based on the measure of correlation *symmetrical uncertainty*, given by

$$SU(X,Y) = 2 \cdot \left(\frac{IG(X|Y)}{H(X) + H(Y)} \right) \tag{5.2}$$

where the function IG represents the *information gain* defined as

$$IG(X|Y) = H(X) - H(X|Y) \tag{5.3}$$

and with H representing the entropy, X representing one of the extracted features, and Y representing the observed class, i.e., the actual secondary task. The entropy of the feature

X is calculated by means of the formula

$$H(X) = -\sum_i P(x_i) \cdot log_2\Big(P(x_i)\Big). \tag{5.4}$$

$H(X|Y)$ describes the entropy of the feature X after observing the corresponding class Y and is defined by

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_j) \cdot log_2\Big(P(x_i|y_j)\Big). \tag{5.5}$$

Class Y has a higher correlation to feature X than to feature Z if the condition

$$IG(X|Y) > IG(Z|Y) \tag{5.6}$$

is satisfied. Usually, a normalization of the information gain IG in Equation (5.2) is performed to handle different occurrences of classes and features in the data. This method comes with the benefit of not being forced to choose the size of the subset a priori. Instead, a relevance threshold γ can be defined to decide if the correlation of any feature and a given class is high enough. For the later evaluations, γ was set to $\gamma = 0.1$.

The head-tracking path, shaded in dark gray in Figure 5.1, is based on the measured head position and head rotation. The calibration step is used to determine the driver's head orientation when looking straight ahead at the road. This is necessary for the later calculation of the head features and for comparability among different test subjects. Note that this calibration step depends only on the applied measurement system and does not conflict with the online-capability of the overall approach. As for the eye-tracking path, the feature extraction step will be skipped here and explained in detail in Subsection 5.2.2.

Finally, the selected eye and head features are merged and used for the training and testing of the classification model. An SVM was employed as classifier, since this type of classifier tends to be robust with regard to the overfitting difficulty due to the use of the regularization principle [116]. Furthermore, a Radial Basis Function (RBF) kernel is applied, because of earlier promising classification results of the combination SVM and RBF kernel [110], [111].

5.2.2 Feature Extraction

The two feature extraction steps of the head- and eye-tracking paths in Figure 5.1 are the main component of the proposed method for DAR and have a significant impact on the classification performance. Hence, in addition to the feature set containing features reported in literature, a set containing novel features especially designed for the automated driving context will be introduced in this subsection. These different sets of features will then be compared to each other based on the data of the pre-study NEBAF (see Subsection 2.4.3) to determine the most promising features in subsection 5.2.3.

In total, 92 eye-based features all taken from literature represent the feature set "*static*". More specifically, 90 features are selected as suggested by Bulling et al. in [110]. These

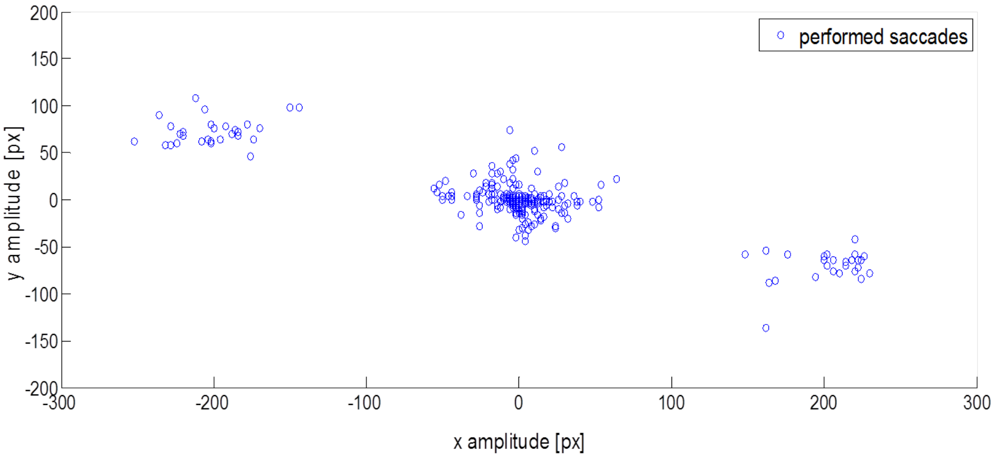


Figure 5.2: Exemplary scatter plot of the EoR gazes. Every point resembles a detected saccade with the horizontal/vertical amount of the amplitude on the x-/y-axis. The scale is in pixels (px), in accordance with the units of the Dikablis eye camera.

features, containing mean, variance, rate, and maximum values, can be separated into four groups: 62 features related to saccades, 5 features derived from fixations, 3 features related to blinks, and 20 wordbook features. The two remaining eye features of this feature set describe the x- and y-coordinate of the centroid of a blink frequency histogram with the actual blink frequency shown on the x-axis and the number of occurrences of the particular frequency shown on the y-axis. This feature was described by Ishimaru et al. in [113] and suggested calculating the blink frequency by shifting a temporal window of specified size over the detected blinks.

These features were already successfully applied for HAR in static lab environments, which explains the label of this feature set. However, a conditionally automated driving scenario can be seen as far more dynamic and distracting than a lab environment. The proof for this assumption of increased dynamic with regard to the eye movement behavior was provided in Chapter 3. Thus, the subjects are expected to diverge, which usually inflicts additional noise and artifacts. However, this altered behavior might contain additional information extractable as novel features. One example for such a phenomenon are the EoR gazes described in Chapter 4 and occurring during secondary tasks when driving in a conditionally automated setting. In Figure 5.2, the recorded gaze behavior of a 5 min interval of such a scenario is plotted. It shows the saccades towards the road (upper-left cluster) and the saccades towards the secondary task (lower-right cluster), exemplarily performed on the area of the center console. In principal, such behavior complicates the goal of detecting the driver activity, since the driver often interrupts the current secondary task leading to eye and head movements that are not related to this task. At the same time however, the number of EoR gazes could correlate with the interruptibility of the task. For example, watching a video does not necessarily require the driver to focus on the screen, resulting in high interruptibility. In contrast, reading a text occupies the visual attention

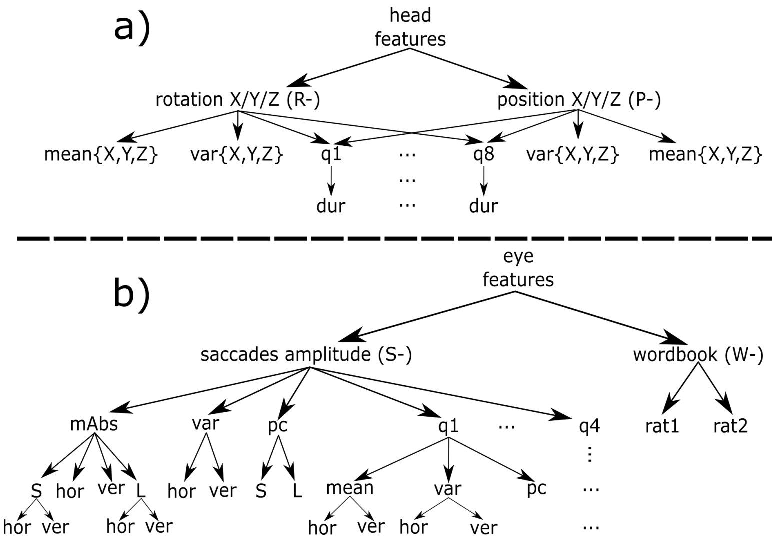


Figure 5.3: Tree structure of the novel head and eye features introduced in this work. mAbs=mean of absolute values, var=variance, hor=horizontal, ver=vertical, S=small, L=large, pc=percentage, qX=quadrant number x, ratX=ratio of words of the size X, dur=duration, B=blink, cent=centroid.

and is usually only interrupted at the end of a sentence or paragraph.

Therefore, this work now examines novel eye and head features introduced to address the behavior of drivers. The set of these 53 features in total will be referred to as the feature set ”*dynamic*”. All these new introduced features are shown in the tree structure in Figure 5.3 along with the appropriate notation. The notation follows the taxonomy introduced by Bulling et al. in [110]. Every leaf node corresponds to an actual feature, while the parent nodes show the dependencies to the different head and eye patterns. Figure 5.3 a) outlines 20 features derived from the head-tracking signal. The mean and variance features are calculated for every position and rotation in the 3-dimensional space. To gain insight into where and for how long the driver's head was directed, the field of view is divided into eight quadrants as shown in Figure 5.4. The inner four quadrants result from the circumstance that the gaze and head direction straight ahead cannot be seen as an exact point but only as a narrow field of view. The size of the inner quadrants was set to 10° in the x- and 5° in the y-direction based on a previous analysis.

Figure 5.3 b) lists the 32 novel eye-based features, where 20 of these features are based on the distribution of driver's saccades in the four outer quadrants $Q1$ to $Q4$ and the remaining twelve features can be seen as an addition to the previously mentioned 92 features. However, in contrast to the features known from literature, the absolute value of the amplitude is used, denoted by *mAbs*, before calculating the mean value. Without using the absolute values, opposite saccade clusters as shown in Figure 5.2, would abrogate one another and information would be lost. The two wordbook features, namely *W-rat1* and *W-rat2*, pursue the idea of improving the classification of secondary tasks involving reading by calculating the ratio between the number of glances to the right and glances to the left. In the case

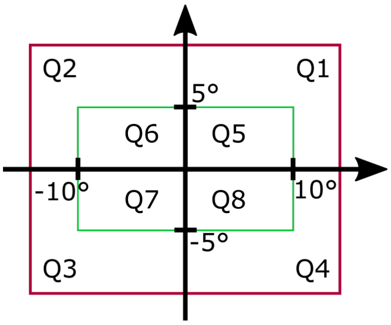


Figure 5.4: Schematic segmentation of the driver's field of view in eight quadrants.

of reading, this ratio should tend towards the glances to the right, since reading typically consist of many small glances in direction of reading. Therefore, all words mapped from saccades with any positive or negative horizontal amount of the amplitude are counted as glances to the right or left, respectively. This feature was calculated for wordbook sizes $l = 1$ and $l = 2$. The remaining features, namely mean, variance, and percentage of the saccades contained in the different quadrants, reflect the distribution of the performed saccades among $Q1$ to $Q4$. Hence, these features determine whether clusters of the EoR gazes exist. The size of the inner quadrants was set to 75 *px* in the x- and 25 *px* in the y-direction based on a previous analysis of scatter plots similar to Figure 5.2.

5.2.3 Evaluation of Static and Dynamic Feature Sets

For the evaluation, only data from 73[1] of the former 85 test subjects of the NEBAF study described in Subsection 2.4.3 could be used due to missing signals from the head- and/or eye-tracking system or erroneous simulations such as traffic freezes for eleven subjects of the experimental group. This is a single subject less than for the evaluation described in Section 3.3 due to one subject with missing head-tracking signals. A One-Against-All Multi-Class SVM classification coupled with a leave-on-out cross-validation method was conducted, i.e. the model was trained with the data samples of 72 test subjects, tested with the samples of the left test subject and this procedure was repeated for every possible combination. This approach was chosen in order to provide as many training samples as possible to cover as many different driver behaviors as possible. Furthermore, this approach ensured that the evaluation was subject-independent, i.e. the model was never trained with any driver data samples used for the testing phase. Since the number of performed secondary tasks varied among the test subjects and the required duration time of some tasks was driver dependent, e.g. because of the reading rate, the number of data samples is not equal for the different tasks. Consequently, training the model with an unbalanced data set was prevented by taking the same number of randomly chosen data samples, more precisely the minimum number of samples among all secondary tasks, of every task.

[1] 41 males/32 females, mean age of 39 years (range 20-60, SD=10)

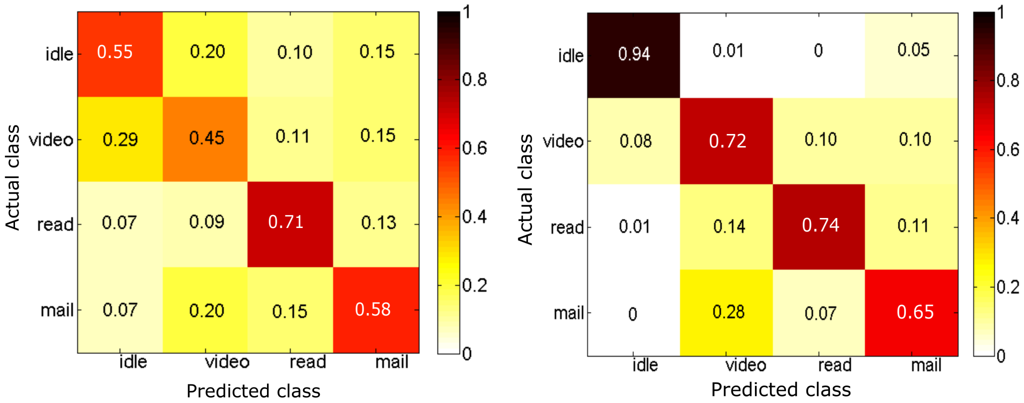


Figure 5.5: Classification results using only the feature set ”*static*” (left) and both feature sets "static" and "dynamic" including the two novel head features (right).

The first step was to apply only the features of the static set in order to determine how these features perform in the context of automated driving. The features were calculated for non-overlapping $90s$ windows. On average, one hour of recorded data per secondary task was used to train the model. The confusion matrix of the classification result is shown on the left side of Figure 5.5. Obviously, the features used are sufficient to distinguish between the different visual tasks and the idle task in the simulated driving scenario. However, with a recall of 0.57% and a precision of 0.5%, the model performs significantly worse than in the known lab environments for which the above features were originally designed. The two histograms on the left part of Figure 5.6 show the features selected by the FCBF. In the upper histogram, the seven most-selected features are seen in the order of the number of selections shaded in gray. Each of these features was selected for at least 75% of the subjects. The cumulative histogram below shows the number of selected features. The average of 26 selected features among the leave-one-out cross-validation is marked with a dotted vertical line.

To improve this classification result, the next step focused on improving the detection of the idle task based on the 20 novel head features as introduced in Section 5.2.2. Three binary classification runs, namely idle task versus video, reading, or writing task, were performed, analyzing the potential of the head features to separate the idle task from the three other visual tasks. In all three cases, the two most relevant and most often selected features were *RP-q4dur* and *RP-q1dur*. The duration for which the head direction stays in one of the two outer quadrants on the right appears to be enough for a reliably classification. All the other features were irrelevant. With respect to these findings, the same three classification runs were repeated using only the *RP-q4dur* and *RP-q1dur* feature. The recall of 0.93% and an average precision value of 0.9% over the three classification results confirmed the identification of two features which can discriminate the idle task from the remaining secondary tasks.

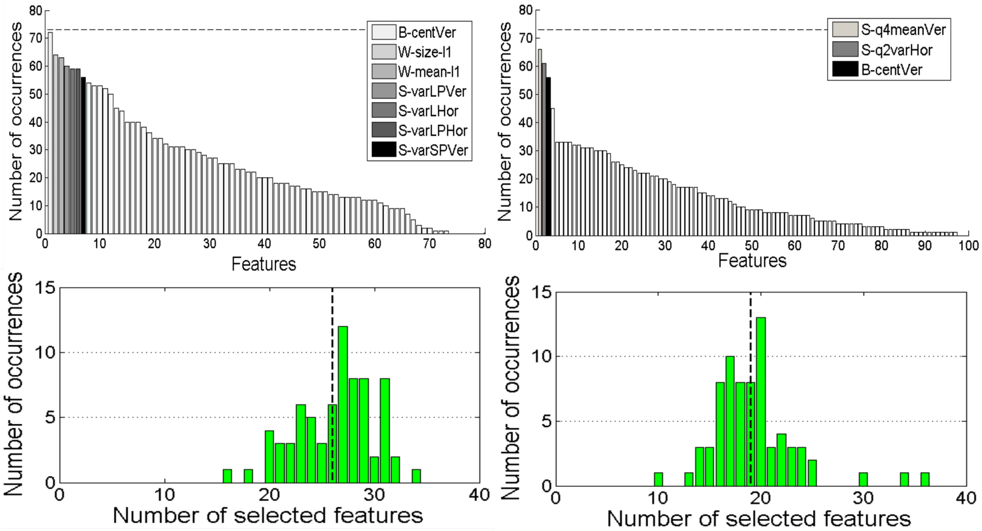


Figure 5.6: Histograms of the selected features for the analysis of the eye features of feature set ”*static*” (left) and for the combination of the feature set "static" with the feature set ”*dynamic*” (right). The upper histograms show the features arranged according to how often they were selected by the FCBF algorithm. The features with the gray shaded bars were selected for more than 75% of the subjects. The lower histograms show the total number of selected features over all subjects, where the dotted line represents the average value.

The final analysis combined the feature sets ”*static*” and ”*dynamic*” with the two a priori extracted head features. In comparison to the first confusion matrix, a significantly improved classification result was obtained as shown in the confusion matrix on the right side of Figure 5.5. Both recall and precision improved by 20% compared to the classification with just the 92 eye features, recall to 0.76% and precision to 0.7%. The best classification is reached by the idle task with a true positive value of 94% while the worst classification result of the mail task can be detected in 65% of cases. The difficulty in detecting the mail task lies in the dual character of the task: it combines segments where the driver is reading and, while writing and focusing on the keypad, segments with unstructured eye movements similar to the video sequences. The high classification rate of the reading task did not increase further since the new features seem to hold no additional information for detecting this task. Furthermore, the upper histogram of Figure 5.6 shows that for the combination of all eye features, a smaller number of features emerges which is used for more than 75% of the subjects. The two best features for this analysis are *S-Q4meanVer* and *S-Q2varHor*, both extracted from the two quadrants directly related to the clusters of saccades emerging from the EoR gaze behavior of the driver. From the histogram below, it can be deduced that the total number of selected features per subject decreases on average by 7 features down to 19 features. Hence, these features seem to contain most relevant information, able of greatly improving the classification result even with a decreased number of features. As presented in Table 5.1, the novel features resulted in a marked improvement

Static feature set				Static & Dynamic feature sets		
Task	**ACC**	**Precision**	**Recall**	**ACC**	**Precision**	**Recall**
idle		0.47	0.55		0.85	0.94
video		0.72	0.45		0.87	0.72
reading		0.6	0.71		0.72	0.74
mail		0.2	0.58		0.35	0.65
Ø	0.53	0.5	0.57	0.77	0.7	0.76

Table 5.1: Summary of the classification results with regard to the applied feature sets. The lowest row contains the averaged results of the different measures.

of the mean classification accuracy[2] (ACC) from 53% to 77%. For later evaluations, the combination of both feature sets will be applied and this approach will be referred to as CIDAR (Chronologically Independent features for Driver-Activity Recognition).

5.3 Scanpath-Based Driver-Activity Recognition

Although the approach described in the previous section showed promising results by incorporating head and eye features and adapting them to automated driving scenarios, it has one major drawback. All features, except for the wordbook features, are calculated by means of a moving time window without capturing any temporal relationship, which means that they cannot contain any information about the chronological order of the patterns detected inside this window. This results in a loss of information, which has a severe negative impact especially when the window size is decreased. Moreover, since this approach is based on raw features which are not further pre-processed, they may include unnecessary or even erroneous information, which could deteriorate the classification performance.

One possible way of considering the chronological order of eye gaze patterns, can be realized by analyzing the visual scanpath of the driver. The assumption for this approach is that similar scanpaths describe the same cognitive activities across multiple subjects. For example, while reading a text from the left to the right, many small saccades to the right are performed before the end of the line is reached and a large saccade to the left is performed to switch to the next line (cf. Figure 5.7 (a) and (c)). This typical pattern is repeated over and over again, which makes it a sufficient indicator for this task. It can be assumed that other secondary tasks also contain such typical patterns. This section is mainly based on the author's journal paper [17].

[2] $Accuracy = (TP+TN)/(TP+FP+FN+TN)$

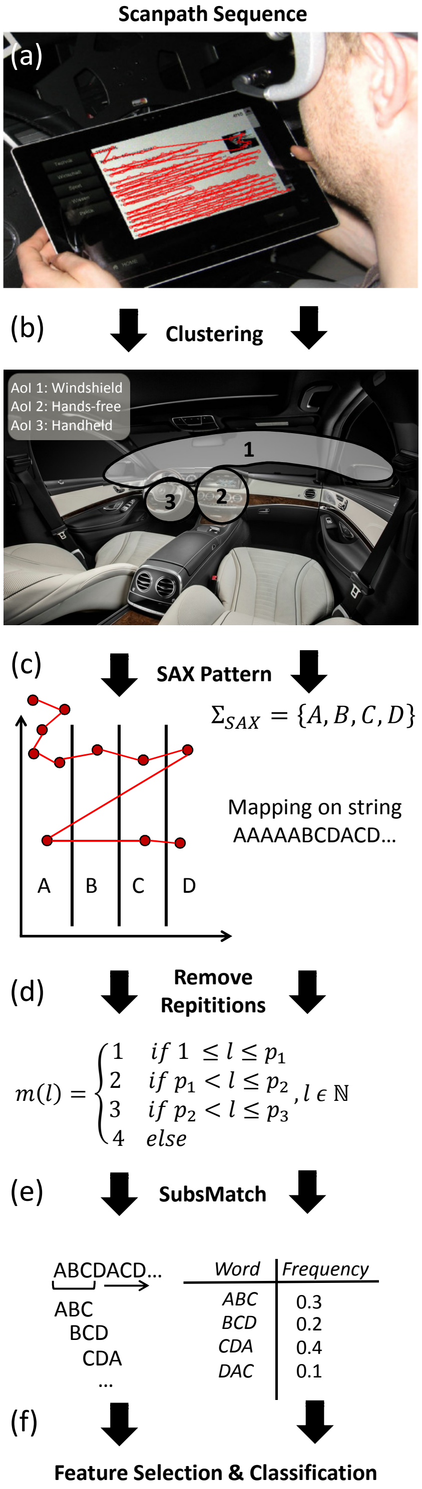


Figure 5.7: Overview of the scanpath-based approach showing the main steps of the algorithm: (a) Recording of the scanpath, (b) Clustering of the AoI, (c) Creating strings over a SAX pattern, (d) Removal of repetitions, (e) Creating tables with frequencies of words according to SubsMatch, (f) selecting the most relevant words and using them for classification.

The method for scanpath comparison as shown in Figure 5.7 is applied to the data of a moving time window similarly to the CIDAR approach. These data sets of the moving window will be referred to as *sequences* with a specified size m_{seq} and step size n_{seq}. The data itself represents the relative gaze direction $\mathbf{\Phi}$ described by the yaw ψ and pitch angle θ of the gaze vector originating at the bridge of the driver's nose. For all samples of each sequence, the algorithm determines the superior AoI which the driver is looking at, e.g. a handheld device or the area of the windshield. This is done by means of the clustering-based approach described in Section 4.3. By shifting a temporal moving window over this signal describing the AoI and performing a majority decision based on the detected AoIs within this window, the approach is able to distinguish between tasks performed on a handheld or hands-free device. Moreover, a driver who is monitoring the AoI describing the *windshield* during automated driving is classified as idle. Data sets classified as idle phases will not be forwarded to the subsequent steps. In subsequent steps, a more subtle detection of the current activity is performed based on a framework for scanpath comparison and a trained SVM.

5.3.1 SAX Patterns

Symbolic aggregate approximation, or simply SAX, is a common method in the field of data mining for mapping time series of arbitrary length on sequences of symbols of a defined alphabet Σ_{SAX} [117]. For example, in Figure 5.7(c) Σ_{SAX} represents a set of four symbols $\{A,B,C,D\}$. In the present approach, the mapping is based on the gaze angles ψ and θ and additionally on patterns, which will be called SAX patterns in the following. Depending on these patterns and the corresponding quantization, different aspects of the gaze behavior will be highlighted, while others will be ignored. This is an especially useful feature for the novel DAR approach, since distinctive eye movement patterns for different secondary tasks can be assumed. Before applying the patterns, $\mathbf{\Phi}$ is high-pass filtered by the mean of the corresponding cluster so that θ and ψ are centered around zero. This high-pass filter allows inter-individual differences to be ignored, e.g., different positions of the tablet or the head direction. In a subsequent step, the difference signal of this high-pass filtered signal is calculated and used as input for the SAX patterns. The vertical SAX pattern was selected as a promising SAX pattern for detecting reading subjects(cf. Figure 5.7(c)). The vertical SAX pattern separates the measured samples in $|\Sigma_{SAX}|$ horizontal areas, e.g., in Figure 5.7(c), the vertical SAX pattern consists of $|\Sigma_{SAX}| = 4$ areas. The width of these areas is determined over quantiles such that they contain the same number of measured samples. This approach was proposed as part of the SubsMatch algorithm by Kübler et al. in [118] to eliminate spatial offsets, e.g., due to different distances to the scene. Since only the yaw angle ψ is considered for the quantization, only horizontal gaze shifts are encoded by this pattern. In the following, the vertical SAX pattern is used in each application of the scanpath-based approach, including the subsequent evaluation. The result of this step is a string of symbols representing the transformed scanpath.

5.3.2 Removal of Repetitive Symbols

In case of only minor or a lack of changes in the gaze direction, data samples are mapped by these patterns on the same symbol for a certain amount of time. Hence, it is necessary to define how to handle many repetitive symbols, especially for high sampling rates. In addition, some SAX patterns increase the probability of repetitions. For example, the vertical SAX pattern ignores vertical eye gazes. Thus, even if huge vertical gaze shifts occur, they will not result in a shift of the currently mapped symbol. The length l describes the number of sample points mapped on the same symbol. Depending on the secondary task, which shall be recognized, the duration can contain crucial information for the later classification or may simply complicate the extraction of the important aspects. Therefore, this approach contains a trade-off between considering and ignoring the length of the repetitive symbols by implementing the function defined in (5.7).

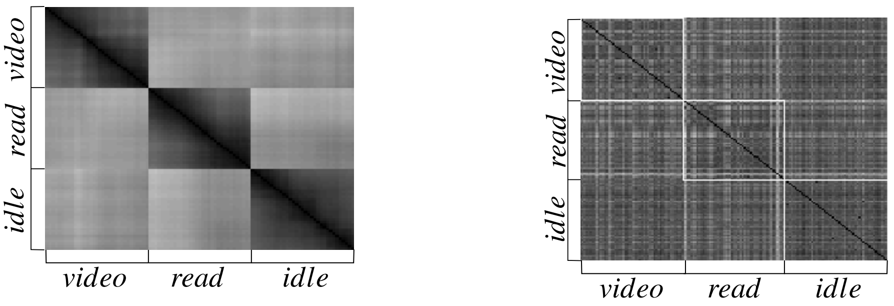
$$m(l) = \begin{cases} 1 & \text{if } 1 \ \leq l \leq p_1 \\ 2 & \text{if } p_1 < l \leq p_2 \\ 3 & \text{if } p_2 < l \leq p_3 \\ 4 & \text{else} \end{cases} ,l \in \mathbb{N} \tag{5.7}$$

By applying the function defined in (5.7), the maximum number of repetitive symbols is reduced to four. Following this approach, the memory consumption of long segments mapped on the same symbol can be significantly reduced and limited, but there is still a coarse distinction of different fixation durations. The values p_i with $i = \{1,2,3\}$ are determined by means of the $(1-16^{1-i})$-th quantile of the length of all segments generated by the vertical SAX pattern of the corresponding subject over all AoIs with at least one repetitive symbol. That means that the number of symbols which need to be removed increases exponentially with the length l of the repetitive segments. In line with the assumption that the entropy decreases for longer repetitive segments, these segments are weighted lower by the function defined in (5.7). This approach was chosen to enable a distinction of segments with usually more than ten repetitive symbols since it is assumed that these segments contain more relevant information than the segments with less repetitive symbols.

5.3.3 SubsMatch

SubsMatch is a framework for comparing scanpaths, which are represented as strings, in dynamic, interactive scenarios [118], [52]. For this purpose, SubsMatch shifts a moving window over the corresponding string generating overlapping substrings, called *words*. These are the same words used for the wordbooks described in Subsection 5.2.1. Afterwards, the frequencies of all existing words of a pre-defined size within each sequence are determined. Each word and the corresponding frequency of occurrence for a given sequence with the length m_{seq} is stored in a normalized hash table. To determine the distance of two sequences, the cumulative, absolute difference between the frequencies of all corresponding words of both hash tables is calculated. Thus, two sequences with almost equal frequencies of the same words are considered as similar. Moreover, the order between words has no impact on the comparison, since there is no weighting of the frequencies of one sequence. Instead, the relevant information on the chronological order of the gaze behavior is contained within the words. Thus, the size of the words m_{word} has to be chosen large enough to cover the relevant gaze behavior of a scanpath. At the same time, it should be noted that the size of the words m_{word} is a critical factor with regard to the memory consumption and run-time of the approach. The number of possible words of the hash table is given by $|\Sigma_{SAX}|^{m_{word}}$ and, therefore, increases exponentially with the size of the words.

To visualize the similarity between hash tables for one and multiple subjects, two distance matrices were calculated with an alphabet size of $|\Sigma_{SAX}| = 9$ symbols, a word size of $m_{word} = 4$ symbols, a sequence length of $m_{seq} = 60$ s, and a step-size of the sequences of $n_{seq} = 1$ s and plotted in Figure 5.8. In Figure 5.8(a), the distance matrix of one subject with three tasks, namely test subject is reading, watching a video, or is being idle, is shown. On the x-axis, a fixed number of tables of each task is used for the calculation of the distances to the same tables plotted on the y-axis. For each calculated distance of two tables, a small box is plotted in an appropriate gray scale. The darker the box the more similar the corresponding tables. Figure 5.8(b) was constructed in the same way. However, for the second distance matrix the tables were randomly drawn out of 42 different subjects. Since the same tables were used on the x- and y-axis for both figures, the dark main diagonal, referring to the high similarity of the corresponding tables, is easy to recognize in both plots. It can be seen that for one subject the tables of each task form a distinctive cluster in the matrix. This indicates that the detected words and, therefore, the viewing patterns are largely similar within the data for one task from one subject. These clusters are no longer recognizable in the lower distance matrix, indicating the inter-individual differences of the viewing patterns. However, the following subsection shows the clusters can be recovered by reducing the dimension of the tables.



(a) Distance matrix of one subject with visible clusters.

(b) Distance matrix of 42 subjects without visible clusters.

Figure 5.8: Distance matrices of the hash tables from example tasks. The gray scale of the small rectangles corresponds with the distance of the corresponding tables of the x- and y- axis. The darker the box the more similar the tables.

5.3.4 Feature Selection and Classification

The tables calculated by means of the SubsMatch algorithm may have a high dimension, depending on the possible number of words given by $|\Sigma_{SAX}|^{m_{word}}$. For such high dimensionality, the probability of overfitting increases, since a tremendous amount of data is necessary for training the classifier. Hence, the size of the tables needs to be reduced. First, all words which never occurred in the tables as well as words with a similar frequency for different secondary tasks are deleted. Second, a feature selection is performed by means of the FCBF (see Subsection 5.2.1) on the remaining words. For that, all frequencies are scaled logarithmically to prevent numerical issues due to the minor frequencies.

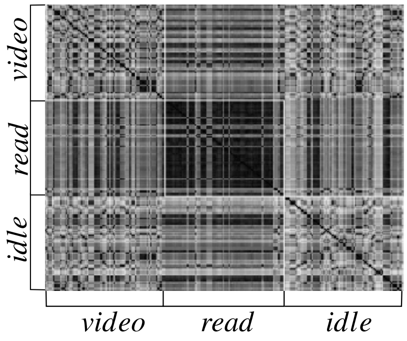


Figure 5.9: Distance matrix of 42 subjects after recovering the clusters using feature selection.

Figure 5.9 shows the same distance matrix as in Figure 5.8(b) but after the performed feature selection. It can be seen that the cluster for the task *read* is recognizable again. Finally, the selected patterns are applied by the classifier to decide which class has to be assigned. In case of various secondary tasks, a cascade of multiple classifiers can be constructed each separating two types of classes. For each step of the classification cascade, a SVM classifier can be trained, or features derived from the AoI cluster can be used directly. For the present work, however, one SVM and the clustering-based approach are sufficient since only three tasks are analyzed. For the SVM a Radial Basis Function kernel was chosen to increase comparability with the approach in Section 5.2.

5.4 Evaluation and Comparison

For the evaluation and comparison of the two approaches described in Sections 5.2 and 5.3, data from 82[3] of the former planned 112 subjects of the KoHAF-study could be applied. For 16 subjects the eye-tracking software froze or could not track the pupil sufficiently, so that the data of these experiments cannot be applied to the two approaches.

[3] 46 males/36 females, mean age of 38 years (range 20-58, SD=11)

In addition, the simulation software of the driving simulator broke down, resulting in the cancelation of ten experiments. Furthermore, the eye tracker suffered mechanical damage after the 56-th subject: The swan-neck mounting of the right eye camera broke while adjusting the camera to the face of the test subject. Hence, only the data of the left eye is used in the following evaluation. The remaining four subjects could not participate, since they did not arrive in time for their scheduled slot. Out of the total number of subjects, 14 subjects belonged to a control group who only performed the task *idle*. These subjects are excluded from the classification of the classes *handheld* versus *hands-free*.

The secondary tasks of each subject were separated into sequences of size m_{seq}. For the evaluation, m_{seq} was varied from 5 s to 90 s with a constant step-size of $n_{seq} = 2\,\text{s}$. The smaller the sequence size, the less information is available for the DAR. Only sequences which can be unambiguously assigned to a single secondary task were used for the evaluation. The ground truth was given by the software running on the supplied tablet, which indicated the currently assigned secondary task and currently opened submenu of the user interface. Note that the subjects were not instructed on how to perform the corresponding tasks in order not to influence their natural behavior. Hence, the subjects sometimes interrupted the secondary task by observing the road or by relaxing for a short moment. Since at the beginning of each task the subjects needed to navigate to the corresponding submenu of the user interface, the first ten seconds of each secondary task were not evaluated. For all evaluations, a leave-one-out cross-validation was performed with a subject-independent testing dataset. That means the SVM was tested with datasets of an unknown subject. A balanced training dataset was generated for each evaluation, so that the training of the classifier would not "prefer" one of the classes. In total, the analyzed simulator data contains about 9 hours of the secondary task *read*, about 7 hours of the secondary task *video*, and about 3 hours of sequences with idle drivers. Furthermore, considering a step-size of 2 s for the calculation of the overlapping sequences by means of a shifting window, about 22 hours of the class *read*, about 17 hours of the class *video*, and about 6.5 hours of sequences of the class *idle* are used for the leave-one-out cross-validation. For each class except for the class *idle*, about 50% of data is performed on a handheld and hands-free device, respectively.

For the sake of simplicity, the terms *Scan* and *CIDAR* will refer to the proposed scanpath-based approach and approach with chronologically independent features, respectively. For CIDAR, all parameters were chosen as described in Section 5.2 and both feature sets ”*static*” and ”*dynamic*” as well as the two best performing head features *RP-q1dur* and *RP-q4dur*. With regard to the selection of the DAR approach parameters based on the driver’s scanpath, the size of the words was set to $m_{word} = 4$ with an alphabet size of $|\Sigma_{SAX}| = 9$, resulting in 6561 possible words.

5.4.1 Handheld vs. Hands-free device

Using CIDAR, a SVM was trained to separate tasks performed on a handheld or hands-free device. Moreover, the clustering approach of Scan was applied for the same task. In Figure 5.10, the F1 score (dashed blue line), representing the harmonic mean of the recall and precision of the binary classification, as well as the accuracy (solid green line) of the leave-one-out cross validation over the different sequence sizes was plotted for both methods. As can be seen, both statistic measures provide similar values for all sequence sizes for both approaches. This indicates that the classification is not performed in favor of one of the two tasks *handheld* or *hands-free*. The F1 score and the accuracy of the classification performance of CIDAR (lines with circular markers) never fall below the 73% mark in Figure 5.10, but show variations for the different sequence sizes. Note that the actual variation of the plot only comprises a range of 7%, i.e. the variations of the measures are quite small. These variations and possible outliers can be ascribed to the directly applied raw features, which are not pre-processed or further abstracted to remove unwanted or erroneous information. CIDAR seems to be a reasonable choice for detecting drivers using handheld or hands-free devices even in case of little information, as long as fluctuations of the performance are tolerable.

In contrast, the clustering approach of Scan (lines with square markers) shows a smooth course for both measures with little to no variations at all between adjacent sequence sizes. This continuous behavior of the classification performance can be ascribed to the use of the continuous gaze direction, which is further pre-processed by performing the majority decision within the moving window. Moreover, Scan is more accurate and has a higher F1 score than the corresponding CIDAR results for each sequence size. In particular, both of these criteria increase by 8% in case of a sequence size of five seconds and even for a sequence size of 30 s. While CIDAR has its peak in the classification performance, Scan still outperforms CIDAR by about 5%. In summary, both approaches are capable of separating tasks performed on handheld or hands-free devices for online- as well as offline-settings. However, Scan should be favored due to the increased classification performance.

5.4.2 Classification of Secondary Tasks

In Figure 5.11, CIDAR and Scan are compared with regard to their classification performance for the three secondary tasks *idle*, *reading*, and *video*. On the one hand, the accuracy and F1 score of CIDAR again vary significantly over the different sequence sizes due to the use of the raw and unfiltered features. However, a tendency towards smaller values can be observed for decreasing sequence sizes for both statistics of this approach. That is why the accuracy and F1 score are only about 65% for a sequence size of 5 s. For larger sequences, especially for 90 s, the approach achieves the highest results. On the other hand, the plots of the accuracy and F1 score of Scan show very similar values and little variation between adjacent sequence sizes similar to the results of Figure 5.10. Moreover, they show continuously decreasing values for smaller sequences. For the shortest sequence sizes, this approach still shows an accuracy and F1 score of about 77%. Both statistics peak at 84% for 50 s sequences, but decrease again for larger sequences.

Table 5.2 summarizes the recall for each classified secondary task of both approaches, while on the other hand Table 5.3 shows the corresponding precision values. The blue-shaded table cells mark the best classification results with regard to the corresponding statistic and highlight once more for which sequence size the approaches perform best. CIDAR shows a high recall for the idle task for all sequence sizes, while at the same time

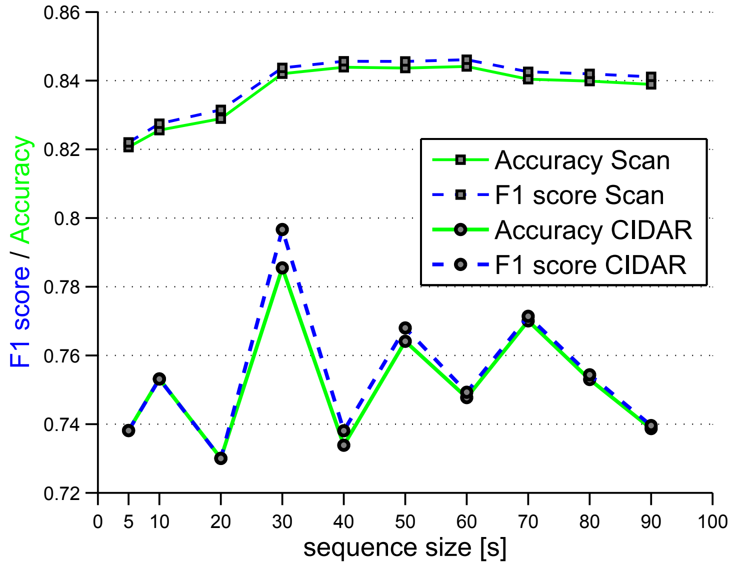


Figure 5.10: The figure shows the accuracy and F1 score for each examined sequence size of the classification task *handheld* vs. *hands-free* performed by CIDAR and Scan.

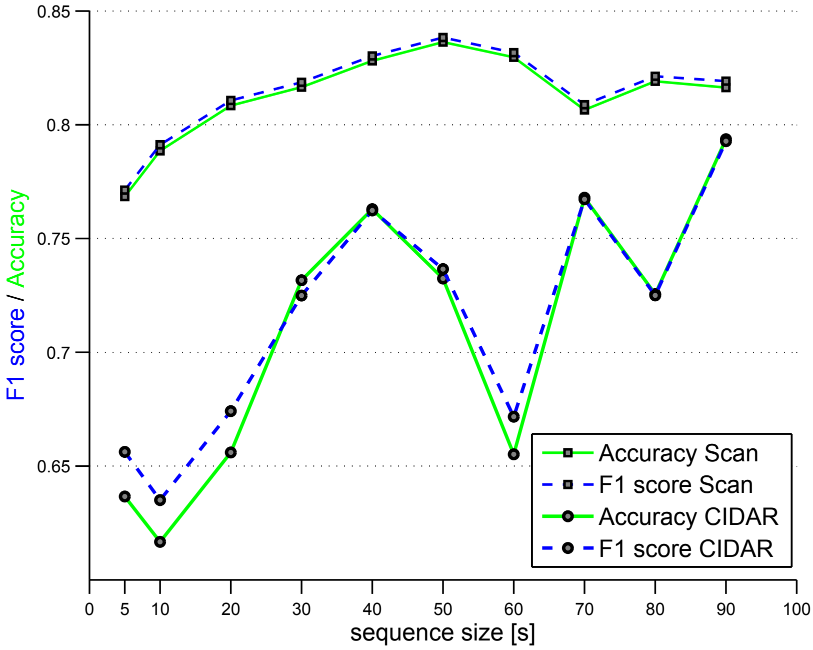


Figure 5.11: Accuracy and F1 score of CIDAR and Scan for each examined sequence size for the classification of the three secondary tasks *read*, *video*, and *idle*.

only the precision for a sequence size of 90 s is at an adequate level of 85%. On the other hand, it can be seen that Scan achieves the same high recall and precision, in detail 91%

Recall of both Approaches						
Size	CIDAR			Scanpath-based		
	Idle	Read	Video	Idle	Read	Video
5 s	0.93	0.62	0.54	0.91	0.72	0.68
10 s	0.92	0.63	0.48	0.91	0.73	0.73
20 s	0.91	0.62	0.60	0.91	0.76	0.75
30 s	0.82	0.73	0.69	0.91	0.78	0.76
40 s	0.92	0.76	0.70	0.91	0.79	0.78
50 s	0.88	0.70	0.69	0.91	0.81	0.79
60 s	0.89	0.61	0.61	0.91	0.80	0.78
70 s	0.89	0.75	0.74	0.91	0.77	0.74
80 s	0.88	0.73	0.65	0.91	0.78	0.76
90 s	0.90	0.80	0.74	0.91	0.75	0.79

Table 5.2: Recall values for each classified task and both approaches calculated for each examined sequence size. The highest values for each task and approach are highlighted in blue.

and 97% for each sequence size. Regarding the recall of the tasks *read* and *video*, CIDAR only exceeds Scan in case of the reading task for a sequence size of 90 s. This remains valid for the precision of the task *video*, where Scan outperforms CIDAR, except for the largest sequence size. However, the precision of the task *read* obtained with Scan remains lower than CIDAR.

All these measures indicate that CIDAR delivers reasonable results only for the largest analyzed sequence size of 90 s. Especially the precision of the task *idle* decreases significantly for all other sequence sizes, which explains the high recall values for this task even for short sequences. Furthermore, this approach never exceeded an accuracy and F1 score of 80% in the described evaluation, which should be possible at least for larger sequences. Based on these findings, the approach is only suited to offline applications and reveals the necessity for an improved, online-capable method. In contrast to this method, Scan shows the benefit of considering the temporal order as well as the use of pre-processed and abstracted features. Similar to the evaluation described in section 5.4.1, the pre-processed features reduce the variations and generate a more continuous behavior of the classification performance. This is done by means of the SAX patterns, which extract only the required information and therefore result in similar pattern sequences for multiple subjects as well. Hence, the algorithm is able to maintain a high classification performance even for short sequence sizes. The classification performance of Scan peaks for a relatively long sequence, but also generates good results for the shortest sequence. That makes this algorithm suitable for both online- and offline-settings. Another reason for the significant improvement is given by the applied clustering-based EoR detection. Scan classifies sequences where the subject was gazing into an AoI *windshield* as a data set of the class *idle*. Since the detection of gazes into the AoI *windshield* is highly accurate and robust, except for the actual detection of short EoR gazes (see evaluation in 4.3.2), and no overlapping

Precision of both Approaches						
Size	CIDAR			Scanpath-based		
	Idle	Read	Video	Idle	Read	Video
5 s	0.49	0.75	0.62	0.97	0.68	0.68
10 s	0.48	0.73	0.59	0.97	0.72	0.69
20 s	0.52	0.80	0.60	0.97	0.75	0.72
30 s	0.55	0.86	0.69	0.97	0.76	0.73
40 s	0.57	0.90	0.73	0.97	0.79	0.74
50 s	0.28	0.88	0.71	0.97	0.80	0.75
60 s	0.52	0.80	0.59	0.97	0.79	0.75
70 s	0.63	0.91	0.69	0.97	0.74	0.72
80 s	0.59	0.85	0.66	0.97	0.77	0.73
90 s	0.85	0.85	0.75	0.97	0.78	0.72

Table 5.3: Precision values for each classified task and both approaches calculated for each examined sequence size. The highest values for each task and approach are highlighted in blue.

with other AoIs occurred in the data, it achieves a remarkable accuracy reflected in the constant high recall and precision of the idle task for all sequence sizes. If the two approaches are compared quantitatively to each other regarding their optimum results for an offline- and online-application, i.e. comparing the results of Scan in case of 50 s and 5 s with the results of CIDAR for 90 s and 5 s, the huge benefit of Scan becomes obvious. While for an offline-application the improvement by Scan with regard to the absolute statistics averaged over the three secondary tasks is quite low $(\varnothing recall \uparrow 2\%, \varnothing precision \uparrow 2\%, \varnothing accuracy \uparrow 2\%, f1score \uparrow 2\%)$, the algorithm shows its impressive benefit in case of an online-setting $(\varnothing recall \uparrow 7\%, \varnothing precision \uparrow 15\%, \varnothing accuracy \uparrow 12\%, f1score \uparrow 11\%)$. Referring to the accuracy of CIDAR for a 5 s sequence, this is a relative increase of about 19%.

5.4.3 Feature Analysis

To analyze the features selected by the FCBF for the final classification of both approaches, the number of features used for the different sequence sizes is plotted in Figure 5.12 and Figure 5.13. For each sequence size, these figures show a bar including the different numbers of selected features over the cross-validation and highlight the median with a star-shaped marker. In Figure 5.12, it is shown that for CIDAR, smaller numbers of selected features no longer appear for sequence lengths from 5 s to 20 s, i.e. the minimum of the first three bars increases. Similarly, there is a slight increase of the median for the three smallest sequence sizes, though it is not significant. This observations indicate that in case of reduced sequence sizes, the smaller numbers of selected features no longer occur and usually more features are necessary to maintain the performance. The number of chosen features for sequence sizes larger than 40 s corresponds to the reported results for the NEBAF data set in Subsection 5.2.3. The bar of the sequence size of 30 s shows a median value of 56 features and ranges from 42 to 62 features. Compared to the values of the remaining bars, this is a significant increase of the median, minimum, and maximum of the plotted bar and shows the possibility of outliers. For Scan, the number of selected features which correspond to a word of $m_{word} = 4$ symbols is plotted in Figure 5.13. The median of the number of selected features remains inside the narrow band between 51 and 56 features for all sequence sizes larger than 10 s. Moreover, each of these larger bars as well as the bar for the shortest sequence size, shows a similar range of the number of selected features, while the median of the 5 s sequence decreases. These findings indicate that most of the possible 6561 words are discarded for the actual classification and that the number of features is kept relatively stable. However, outliers, e.g. for a sequence size of 10 s, can occur and result in the selection of far more features than usual. Nevertheless, this significant change in the number of features has no obvious impact on the continuous behavior of the classification performance (cf. Figure 5.11). In summary, CIDAR selects on average about half as many features as Scan for the DAR. Further, note that Scan provides up to 6561 features to the feature selection step whereas CIDAR only provides 125 eye and two head features. Although most of Scan's features are discarded, this large number of words seems to be more suited for covering the individual requirements of the DAR. The number of selected features of both approaches ranges within a comparably constant interval for different sequence sizes but is at the same time prone to outliers. An interesting difference between these two approaches is the behavior for varying sequence sizes. While for CIDAR the number of features seems to increase for shorter sequences, Scan shows the lowest median of selected features for the shortest sequence of 5 s. This means that for CIDAR, more features are assumed as relevant for a decreased classification performance whereas for Scan, as expected, more features become useless. This indicates that some of CIDAR's features contain deceptive information which does not correlate with the secondary task.

For a more detailed view on the applied features, Figure 5.14 and Figure 5.15 show the top ten features for a 5 s sequence size. The ranking was calculated by summarizing the occurrences of each feature weighted by the respective feature rank over all test subjects of the performed leave-one-out cross-validation. Figure 5.14 shows that for CIDAR more than half of the plotted features were selected by the FCBF for at least 50% of the subjects. This demonstrates that these features truly contain patterns and information suitable for the classification task over multiple subjects. Moreover, these features usually show a narrow range and a high rank (low value on the y-axis) as median rank, except for the feature *R-meanZ*. For the head features *RP-q4dur* and *R-meanZ*, the actual overvalue can be easily seen and interpreted. For example, the top ranked feature *RP-q4dur*, chosen for each test subject as most relevant feature, describes the duration the head pose remained in the fourth quadrant of the field of view, which corresponds to the area of the mounted display for the hands-free variant. Again, these results correspond to the results in Subsection 5.2.3, since the feature *RP-q4dur* was also reported to be the most meaningful head feature in the classification task.

Figure 5.15 shows the same analysis for Scan. Note that Scan uses features on a meta level. In other words, the algorithm defines the features during runtime and according to the individual AoIs and the calculated quantiles of the areas of the SAX pattern. Due to

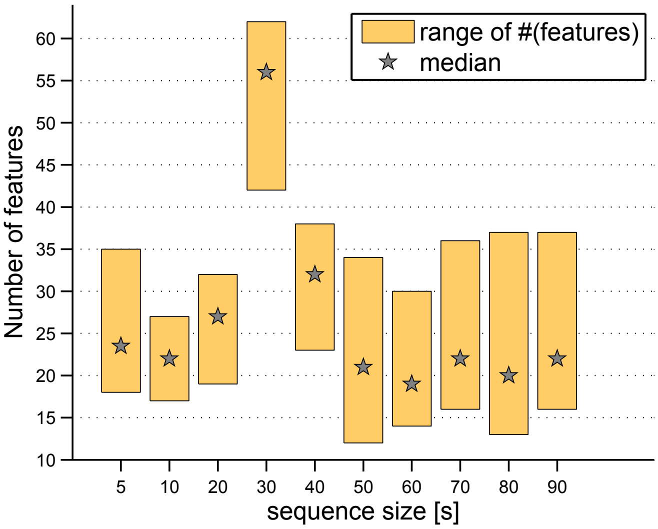


Figure 5.12: Number of selected features for each sequence size over all test subjects for CIDAR. Each bar visualizes the range given by the minimum and maximum number of selected features. The median of the selected number of features is given by the star-shaped marker

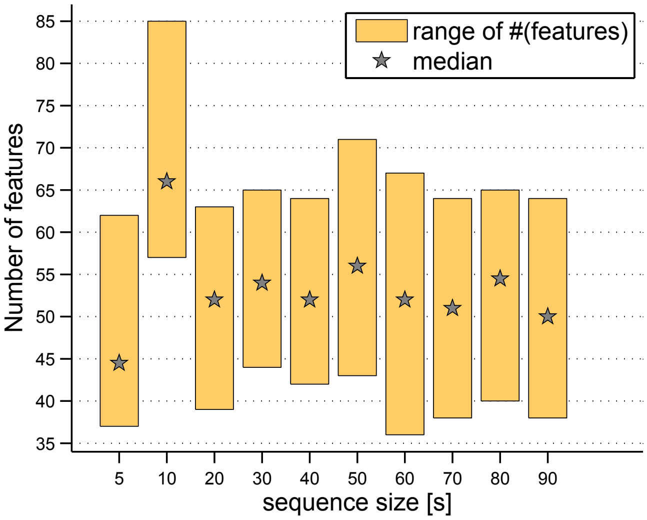


Figure 5.13: Number of selected features for each sequence size over all subjects for Scan. Each bar visualizes the range given by the minimum and maximum number of selected features. The median of the selected number of features is given by the star-shaped marker.

this individual distribution of the gazes, the feature *A-B-C-D* of subject A usually refers to different areas of the SAX pattern than for subject B. However, the spatial order of the areas of one SAX pattern remains the same even for multiple subjects. Since only the vertical SAX pattern with the alphabet $\Sigma_{SAX} = \{A,B,C,D,E,F,G,H,I\}$ was used in this evaluation, the leftmost area of this pattern was assigned to symbol A, the adjacent right

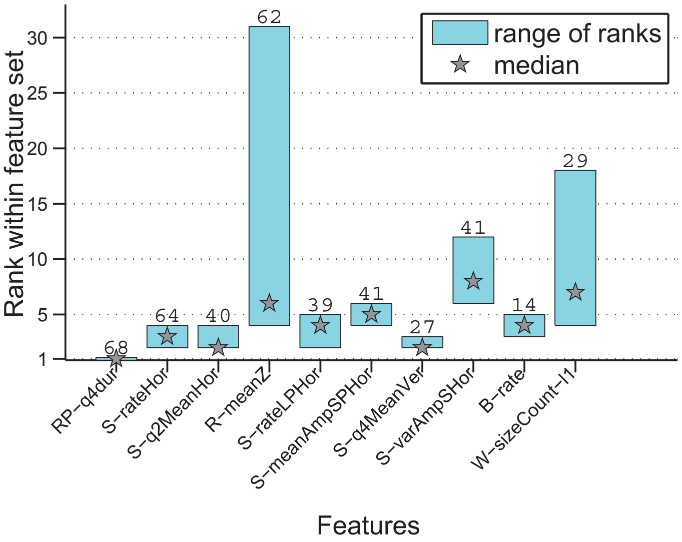


Figure 5.14: The selected features of the FCBF for a 5 s sequence are plotted according to their overall ranking from left to right along the x-axis for CIDAR. Along the y-axis, the rank of the features is plotted, with a lower value representing a higher ranking. The value on the top of each bar gives the total number of times the feature was selected for the leave-one-out cross-validation. The range of the bar is given by the minimum and maximum rank of the corresponding feature, while the star-shaped marker shows the median rank.

area received symbol B, and this continues up to the rightmost area with symbol I. The features listed in Figure 5.15 contain relevant information about the desired patterns that are typical for the corresponding secondary tasks. For example, the top ranked feature $B-E-G-A$ was selected for nearly all 82 subjects as the most relevant of all features. This feature obviously correlates with the expected scanpath while reading, i.e. small steps from the left to the right areas of the SAX pattern with a big step back to the left side of the pattern. Moreover, four of the shown features, including some features with a similar reading pattern, were selected for over 50% of the subjects. This statistic further underpins the assumption that the selected feature really contain relevant information for the later classification, instead of just being randomly selected. Features shown in Figure 5.15, which do not resemble typical reading patterns, could be the result of scanpaths usually performed during the secondary task *video*. To confirm this hypothesis, additional studies need to be conducted. In summary, the top-ranked features of both approaches were chosen for most of the subjects of the cross-validation, thus indicating the relevance of these features. However, it appears that CIDAR contains few features with highest relevance (ranked for most subjects as extremely relevant) whereas Scan makes use of even fewer highly relevant but more somewhat relevant (ranked by some subjects as relevant) features. This explanation corresponds with the findings regarding the number of selected features.

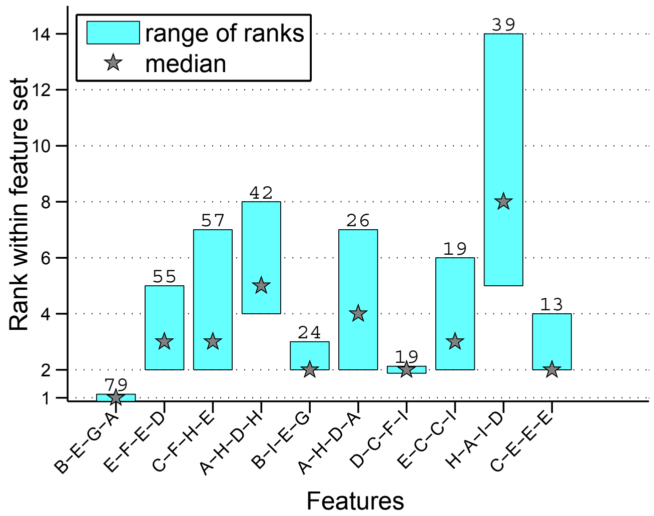


Figure 5.15: The selected features by the FCBF for a 5 s sequence are plotted according to their overall ranking from left to right along the x-axis for Scan. The rank of the features is plotted along the y-axis, with a lower value representing a higher ranking. The value at the top of each bar gives the total number of times the feature was selected for the leave-one-out cross-validation. The range of the bar is given by the minimum and maximum rank of the corresponding feature, while the star-shaped marker shows the median rank.

5.5 Driver-Activity Recognition online in the Vehicle

The scanpath-based DAR was selected for transfer to the testing vehicle because it exceeds the algorithm based on chronologically independent features in terms of classification performance even for smaller sequence sizes. This section describes how this approach is actually implemented to run online and autarkic in a vehicle without any manual input. Especially the calculation of the quantiles for the SAX patterns, the memory- and runtime-critical behavior of the SubsMatch approach, and the final classification step have to be investigated with regard to their applicability to online scenarios. Similar to the EoR detection of the previous Chapter 4, the performance of the modified approach for DAR is evaluated applying a near-to-production driver camera and realistic conditionally automated driving scenarios to show the applicability and the degree of mature in Subsection 5.5.4.

5.5.1 Estimating Quantiles in Online Scenarios

The vertical SAX pattern is used for the evaluation performed in Section 5.4. This pattern determines the size of the different areas used for mapping the gaze direction on the symbols of the defined alphabet by calculating equally sized quantiles. The quantiles can be easily calculated offline since the complete recorded scanpath of each subject is available. However, this approach is not suitable for online scenarios providing only parts of the recorded gaze direction at a time. Moreover, online applications have to deal with limited

memory capacity preventing the storing of large amounts of data sets. Hence, an approach is needed to replace the exact calculation of the quantiles by a sufficient estimation.
As presented in Subsection 4.4.1, quantiles can be estimated online following the approach of Galley et al. [93]. This approach uses a sample mean for estimating the moving average over a defined window size. The moving average represents the estimation for the median, i.e. the 50% quantile. Hence, a first estimation of the median requires at least enough samples to fill the defined window. As soon as a first estimation of the 50% quantile is available, each of the following samples is compared to this value before it is used to update the median. If the current sample exceeds the 50% quantile, this value is also used to estimate the 75% quantile by applying another sample mean estimator. Similar, the 25% quantile is estimated based on each value deceeding the median. Thus, this approach estimates the different quantiles by calculating a moving average of the remaining samples. For example, Figure 5.16 visualizes a tree-structure including the different layers generated by performing three bisections. The variables $\hat{q}_n^{(Q)}$ in Figure 5.16 describe the estimation of the Q-th quantile. The bisection method can be repeated as often as desired. However, the more quantiles are estimated the fewer samples are used for the calculation of each value, which might result in unreliable estimations.

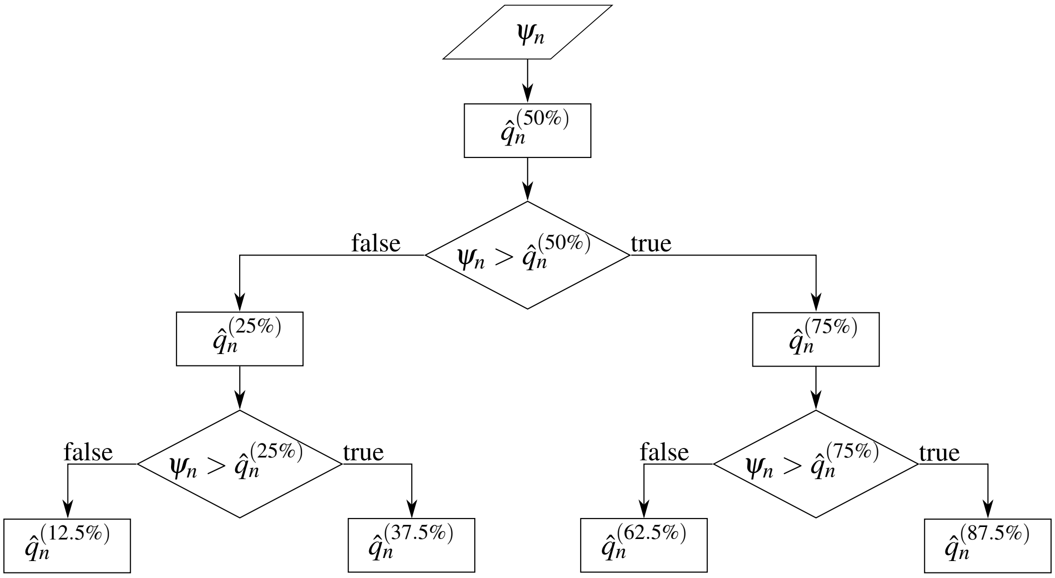


Figure 5.16: Tree-structure of the approach for estimating quantiles of the yaw angle of the driver's gaze in an online-fashion based on bisection.

For the modification of the approach to online scenarios, eight rather than the original nine quantiles of the offline variant were used due to the symmetrical bisection method. Hence, the alphabet size equals $|\Sigma_{SAX}| = 8$ symbols. In a first step, the original estimation approach as described by Galley et al. in [93] was applied to samples of a gaze direction recorded for research purpose. As shown in Figure 5.17, the original approach performs adequately for samples of a gaze direction including only small gaze shifts and an almost static moving average, e.g., reading scenarios. In that case, the distribution of the quantiles is almost equal and corresponds to the actual quantiles calculated offline.

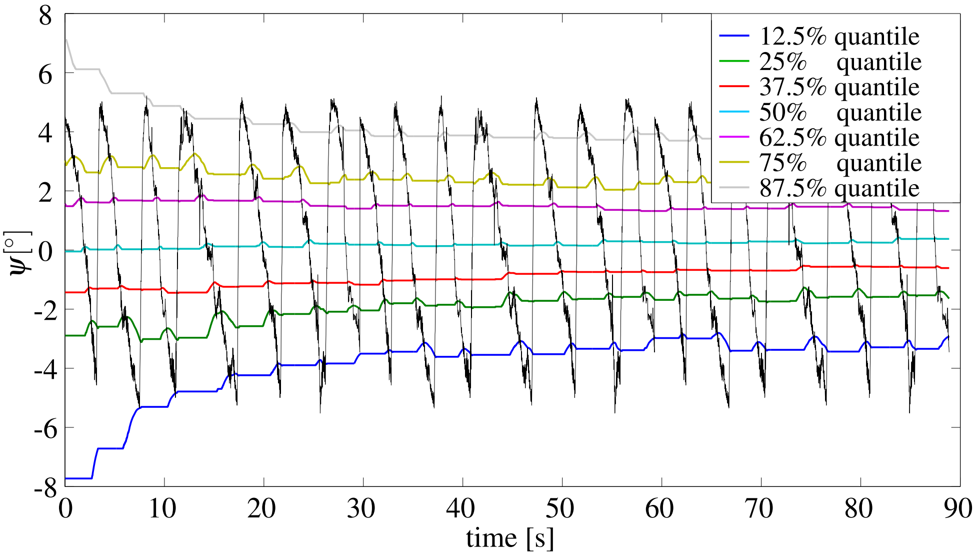


Figure 5.17: For static scanpaths, e.g., without large gaze shifts, the estimation approach proposed by Galley et al. in [93] sufficiently calculates the searched quantiles. All quantiles are equally adapted to the gaze estimation after a short period of time.

However, conditionally automated driving scenarios usually include large gaze shifts to different AoIs and clusters with a dynamic moving average, e.g., for handheld devices. Such phenomena in combination with multiple bisections might cause permutation between adjacent quantiles. Figure 5.18 visualizes an example in which the 25% and 12.5% quantiles falsely exceed the 37.5% quantile after a certain time interval. For the example in Figure 5.18, the difference between the initial quantiles and the actual signal of the horizontal gaze direction is quite high. The reason for such behavior could be focus on a different AoI at the beginning of a secondary task. Note that the estimation of the moving average for the estimation of the different quantiles might be slow depending on the chosen window size. Hence, the algorithm needs some time until it adapts to the gaze direction. Furthermore, not all quantiles are calculated simultaneously. As shown in Figure 5.16, the calculation of a quantile of the lower layers depends on the calculation of the quantiles of the higher levels. As a result, the 50%, 75%, and 87.5% quantiles are learned at the beginning of Figure 5.18 whereas the 62.5% quantile is not updated until the value of the 75% quantile is adapted to the large initial difference. Only then is the horizontal gaze estimation below this quantile and the 62.5% quantile is updated. This asynchronous procedure leads to the false permutation of the quantiles and results in the scenario where the 37.5% quantile is not adapted at all.

To accommodate this behavior, the quantiles are limited by the surrounding quantiles so that no permutation is possible. The modified behavior of the estimation for the same

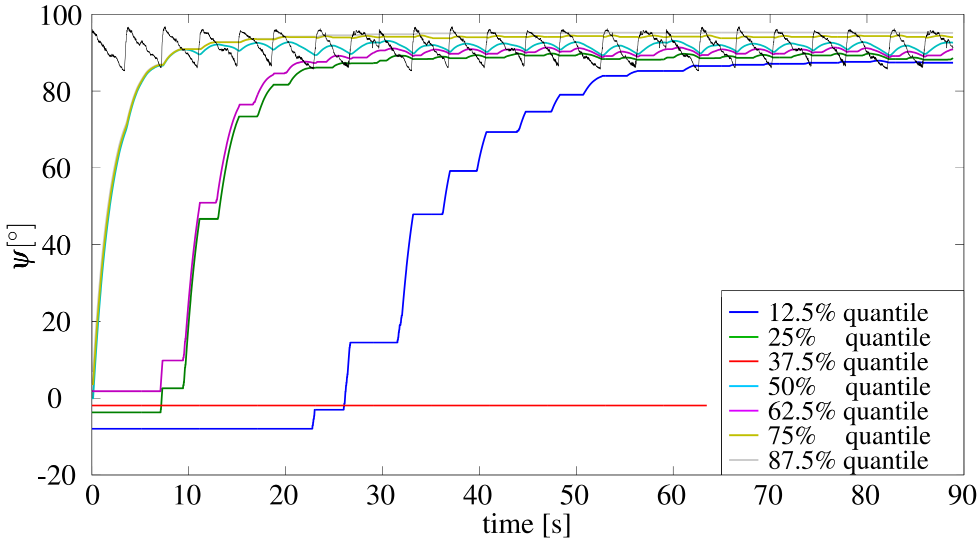


Figure 5.18: The approach for estimating the quantiles are applied to a dynamic scenario in which the initial quantiles show a large offset to the horizontal gaze direction. This could be the result of a large gaze shift while reading on a handheld device. The adaption of the quantiles is slow and an incorrect permutation between the 12.5%, 25%, and the 37.5% quantiles occurs.

example with the large initial difference is shown in Figure 5.19. By limiting the quantiles over the adjacent quantiles, the quantiles are estimated in a given order. At first, the 87.5% is estimated resulting in the adaption of the gray curve in Figure 5.19. After that, the 75% quantile is updated, followed by the 62.5% quantile and so on. The cascading of the calculation prevents the permutation of the quantiles. However, this algorithm further increases the duration of the adaption process since the quantiles are no longer calculated simultaneously. Hence, this algorithm prevents the permutation of the quantiles but shows an even slower adaption process than the previous variant. In summary, the approach is suited only for long scanpaths without any significant differences of the average gaze estimation.

To handle dynamic scenarios and to increase the adaption process, the described concept of estimating the quantiles is modified by a priori estimated, static quantiles. In a first step, the 50% quantile is estimated by a sample mean estimator equal to the variants described above. In contrast, all the remaining quantiles are defined by the distance

$$\Delta_{ref}^{(Q)} = q_{ref}^{(Q)} - q_{ref}^{(50\%)} \quad \text{with} \quad Q \in \{12.5\%, 25\%, 37.5\%, 62.5\%, 75\%, 87.5\%\} \tag{5.8}$$

which represents a constant value. The parameters $q_{ref}^{(Q)}$ and $q_{ref}^{(50)}$ represent averaged reference values and are learned a priori based on a given data set with multiple drivers which were recorded while driving manually and performing secondary tasks in the vehicle. While analyzing these quantiles, it was found that the distance between adjacent

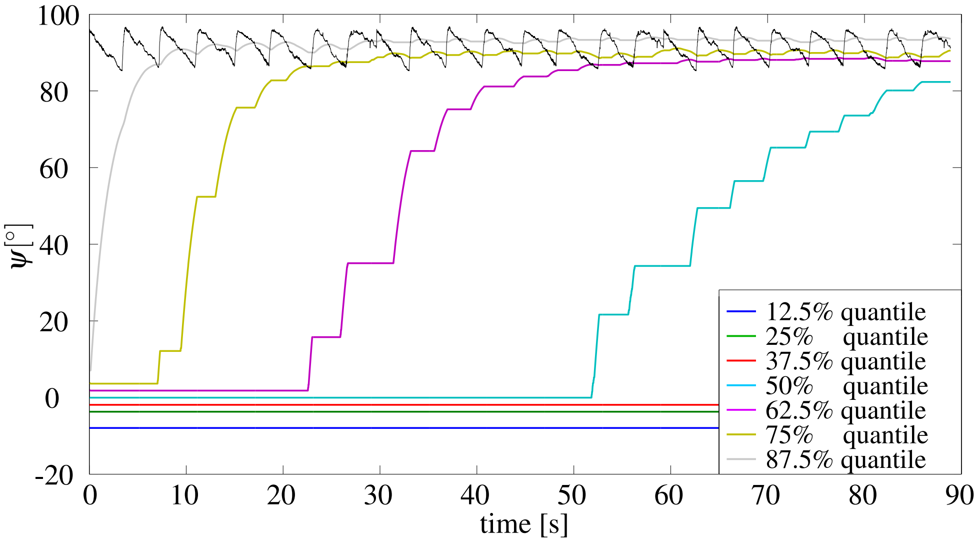


Figure 5.19: The modified approach for estimating the quantiles is applied to a dynamic scenario in which the initial quantiles show a large offset to the horizontal gaze direction. This could be the result of a large gaze shift while reading on a handheld device. By means of the limitation of the quantiles, no incorrect permutation occurs. However, the speed of the adaption process is reduced significantly.

quantiles shows only small inter-individual variations. Instead, significant variations occur only for the absolute positioning of the quantiles due to different head poses, seating positions, body size, or varying AoIs. The constant distance $\Delta_{ref}^{(Q)}$ is applied to generate the estimations of the quantiles of the current driver over

$$\hat{q}_n^{(Q)} = |q_n^{(50\%)} - \Delta_{ref}^{(Q)}|. \tag{5.9}$$

According to this approach, only the 50% quantile has to be estimated for the current driver while the remaining quantiles are defined by their static distance by equation (5.9). Figure 5.20 shows how this approach performs for the driving sequence presented in the previous examples. Since only the 50% quantile is estimated, no false permutations of the quantiles occur and the adaption process is significantly faster. Although the difference between the initial starting position of the quantiles and the actual gaze estimation is as large as in the previous plots, the duration of the adaption process is less than ten seconds. For smaller differences, which are far more common during dynamic driving scenarios, the adaption would be even faster. Moreover, the a priori calculated distances between adjacent quantiles separate the gaze estimation in reasonable areas for the SAX pattern. In summary, the last described variation to estimate the quantiles for the generation of the SAX patterns is the most promising approach with regard to an online application and will be included in our online model of the scanpath-based approach.

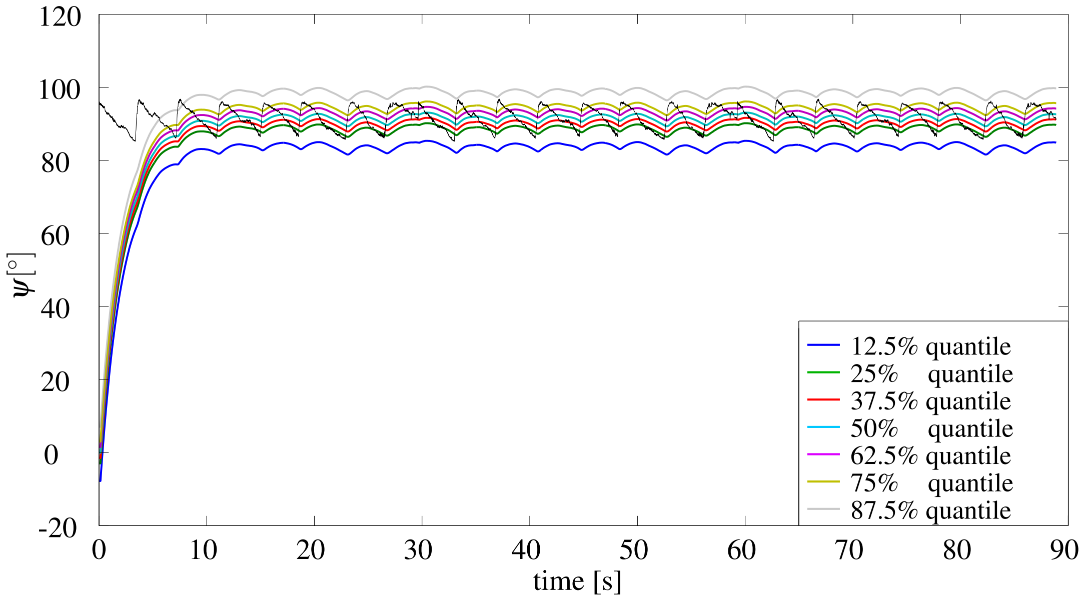


Figure 5.20: A variant of the approach for estimating the quantiles is applied on the same dynamic scenario as for the previous figures and variants. Due to the a priori learned distances $\Delta_{ref}^{(Q)}$ of adjacent quantiles, only the 50% quantile is estimated to adapt the quantiles to the absolute position of the scanpath. Hence, the speed of the adaption process can be improved significantly.

5.5.2 Modifying SubsMatch for In-Vehicle Applications

To extract information on the recorded gaze direction, the SubsMatch approach described in Subsection 5.3.3 is implemented in the vehicle. In a first step, this approach uses the estimation of the quantiles introduced in the previous Subsection 5.5.1 to generate the vertical SAX pattern. This pattern was also chosen for the offline evaluation in Section 5.4. To further maintain comparability between the off- and online variants of the DAR approach, the alphabet size $|\Sigma_{SAX}|$ should be similar. However, the estimation of the quantiles is based on the bisection method and, therefore, the alphabet size $|\Sigma_{SAX}|$ has to be an even number. Hence, the alphabet size of the online variant for the DAR was set to $|\Sigma_{SAX}| = 8$. Based on the applied SAX pattern, a scanpath of a defined sequence size is mapped on a string consisting of the defined symbols of the alphabet. The smaller the sequence size m_{seq}, the faster the recognition of the driver activity. In a second step, words are calculated by shifting a moving window over the generated string of the sequence. The size of the words and the step size of the sequences are set to $m_{word} = 4$ symbols and $n_{seq} = 1$ s equal to the offline variant. Hence, the size of the hash table, i.e. the number of existing words, equals $|\Sigma_{SAX}|^{m_{word}} = 4096$. For each occurring word, the corresponding value is searched in the table and the frequency is updated.

If a sequence size of $m_{seq} = 5$ s is assumed for a sampling rate of 50 Hz, the sequence size equals 250 samples. In general, this is a relative short sequence which exacerbates the detection of the driver's secondary task based on the recorded gaze behavior even for a human observer. Although a short time period, the number of comparisons becomes unacceptably high for the worst-case scenarios. The number of comparisons is given by

$$\left(\left\lfloor \frac{m_{seq}}{n_{seq}} \right\rfloor - m_{word} + 1 \right) \cdot \left(|\Sigma_{SAX}|^{m_{word}} \right) \tag{5.10}$$

which excludes the fragmentary windows at the end of the sequence. In the worst-case scenario, an algorithm which simply compares each word of a sequence with the complete hash table would perform 1011712 comparisons for each step of the sequence. Such a huge number of comparisons would preclude an online-capable implementation of the scanpath-based approach since all these comparisons would have to be performed during each 0.02 s time sample.

To decrease the number of comparisons to an appropriate level, the total number of unique words in the hash table is reduced. This reduction is based on an analysis of hash tables extracted from exemplary data sets. All words which occurred less than a pre-defined number of times in these reference hash tables represent irrelevant features which do not contain any relevant information and, therefore, have to be removed. In a subsequent step, words with significantly higher frequency than the average frequency of the hash table are discarded. These are words which consist mainly of repetitions of symbols such as *AAAA* or *BBBB*. However, these words correlate with the fixation duration instead of with the performed secondary task. Moreover, their high frequency reduces the overall frequencies of the remaining words. Depending on the size of alphabet Σ_{SAX} and the sampling rate, the length of repeated symbols is influenced. For the offline variant of the scanpath-based approach, the number of repetitions was reduced based on equation (5.7) of Subsection 5.3.2. However, the correct calculation of the necessary quantiles for equation (5.7) is not suited to online applications as discussed above. This is why the words which completely consist of repetitions are discarded a priori. Finally, the remaining words are used as input for the FCBF algorithm resulting in a hash table for the later classification with only 28 unique words.

Besides the quantitative conditions of the hash table, the search strategy to locate the correct words and update the corresponding frequencies in the hash table should be optimized. For the above calculation of the worst-case scenario, a simple linear approach with a complexity of $O(n)$ was applied where the hash table is searched for each word of a sequence whenever a new symbol arrives. That means that for the worst-case scenario the complete hash table has to be searched since the wanted word is at the bottom of the table. Note that the location of each occurring word of the sequence has to be searched in the table. Moreover, the moving window to determine the occurring words within a given sequence cannot simply be shifted over this sequence as for the offline variant. The reason for that is the continuous flow of input values and the necessity for defined values for each connection of the Simulink model. In detail, the offline variant of the scanpath-based DAR knows all values of the recorded data set a priori and, thus, is able to filter all invalid sample points and replace them through interpolation. In addition, the moving window which determines the sequences of the size m_{seq}[4] can be filled completely at the beginning of the evaluation as long as the number of available data points is larger than m_{seq}, i.e. the first sequence is filled with 250 valid symbols right from the beginning of the algorithm. On the other hand, the online Simulink model receives one input value, i.e. the next estimated point of the driver's gaze, per sample and then validates this data point. That means that for at least the first 250 samples no complete sequence of size m_{seq} is available. Further, if a value is invalid, e.g., because the camera was not able to detect the driver's eyes, the model still has to occupy each connection between function blocks of the model with a concrete value, since this value cannot simply be discarded. For that case, a default error value which would be integrated in the hash table could be sent instead, deteriorating the classification.

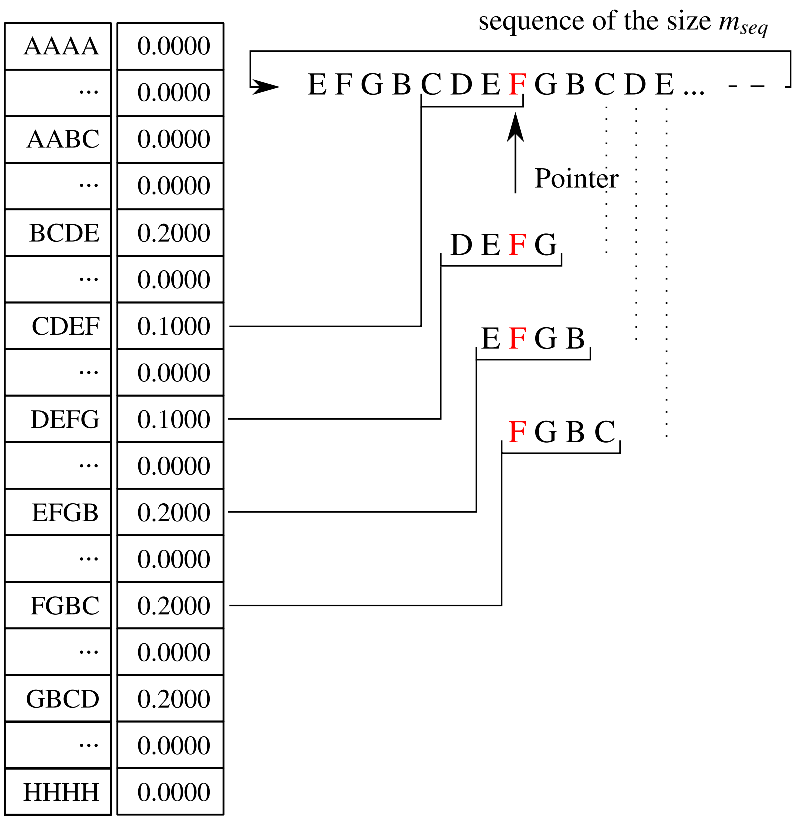
To solve the described problems, the hash table is initially filled with defined words, e.g., *AAAA*, which will be removed by the feature selection described above before the classification step. Hence, for at least m_{seq} frames the classification is based on incorrect values and cannot be trusted. Since for online applications only a limited amount of memory is available, the sequences are dynamically stored in ring buffers. For the ring buffer, a pointer is defined referencing to the currently changed symbol. If the input of the model is a valid data point, both the corresponding symbol and the location which the pointer is referencing are updated. In the example of Figure 5.21, this updated symbol is the colored symbol ”F” referenced by the pointer. In the trivial approach described above, the new updated symbol ”F” requires an update of the complete hash table by determining each word of the sequence and their corresponding position in the table. However, in Figure 5.21 it is shown that the updated symbol only affects the words including this symbol based on the word size m_{word}. In this example, the word size was set to $m_{word} = 4$, which means that only four words of the complete sequence are influenced by the updated symbol. Thus, the number of words which has to be updated for each step is reduced to m_{word} and becomes constant with $O(1)$. Furthermore, the pointer enables the model to ignore invalid values. In case of an invalid value, the complete sequence including the pointer of the sample step $n-1$ is copied to the sample step n. As a result, the sequence of symbols, the hash table, and the output of the model remain unchanged because the pointer is still referencing the correct symbol.
Based on the reduction of the hash table size and the reference mechanism of the pointer, the worst-case scenario is reduced to $224\frac{comparisons}{frame}$, which equals a reduction of about 98% compared to the previous mentioned worst case.

5.5.3 Modifying the Classification Step

The process of classification of the four classes *non-automated driving*, *being idle*, *reading*, and *watching a video* is shown in Figure 5.22. Based on the binary trigger AF[5] it can be determined in a first step if the subject is driving manually. Such a trigger signal has

[4] In the following the sequence size is exemplarily set to $m_{seq} = 250$ frames.

[5] Automated Function



AAAA	0.0000
···	0.0000
AABC	0.0000
···	0.0000
BCDE	0.2000
···	0.0000
CDEF	0.1000
···	0.0000
DEFG	0.1000
···	0.0000
EFGB	0.2000
···	0.0000
FGBC	0.2000
···	0.0000
GBCD	0.2000
···	0.0000
HHHH	0.0000

sequence of the size m_{seq}
E F G B C D E F G B C D E ... - -
Pointer
D E F G
E F G B
F G B C

Figure 5.21: Visualization of the search and update strategy of the hash-table based on a pointer referencing the latest sample point. Only the symbols within the window range m_{word} have to be updated.

to be available in future vehicles with automated driving functions due to legal regulations and enables a classification accuracy of 100% between driving in a non- or automated scenario. The online model of the scanpath-based DAR is provided with input data only if the driver activated the automated driving function. As described in the previous subsection, invalid data samples cause the model to ignore the input and to keep the hash table and output constant. The model of the scanpath-based approach is combined with the model for detecting EoR gazes introduced in Section 4.5. Each time, an EoR gaze is performed the model is stopped in a similar way as for invalid data samples. For valid input samples, the model updates the hash table as described above. For each sample, the updated hash table is compared to a reference table learned a priori with data sets of multiple subjects reading long texts or watching videos. The comparison of step n is done by calculating the Euclidean distance d_n between the current and learned hash table. A difference signal d is plotted in Figure 5.23 for a complete test drive of the study described in Subsection 2.4.5. As can be seen, the distance value decreases quickly to values smaller than 0.4 and increases again when the driver resumes the driving task or is simply idle.

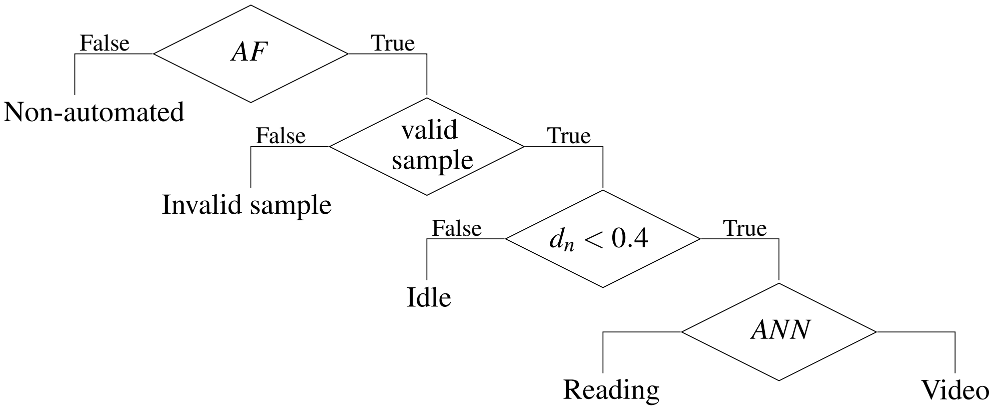


Figure 5.22: Classification process visualized as a cascading decision tree. The variable *AF* represents a binary trigger describing whether the vehicle is driving in an automated mode. Only valid samples are checked for the performance of a secondary task and assigned to the reading or video class.

For the secondary tasks *reading* and *video*, no difference can be recognized in the distance d. Hence, this approach is only suited for separating idle drivers from drivers performing any kind of task. Note that the threshold 0.4 represents half of the maximum Euclidean distance and is an experimental value which generates convenient results. Further tuning of this parameter will most likely not yield a significant improvement in the approach.

To separate reading drivers from drivers watching a video, an SVM was applied in the offline variant presented in Section 5.3. Since the model is implemented in Simulink, there was no toolbox for simulating an SVM in the block diagram environment available. However, Simulink provides an alternative with the *Neural Network Toolbox* enabling the training and application of artificial neural network (ANN) classifiers[6]. Compared to kernelized SVMs which are non-parametric models, the number of parameters for the ANN is fixed (parametric model). As a consequence, the number of parameters of an SVM, i.e. the support vectors, may increase for large amounts of training sets while the number of parameters of the ANN remains constant. The parameters which have to be selected for the ANN of the Simulink toolbox are the number of input neurons (equals the size of the hash table), the number of output neurons (number of secondary tasks), and the number of hidden layers. As a rule of thumb, the number of hidden layers should lie between the number of input and output neurons. The values of the output neurons after the processing of the ANN are scalars of the interval $[0,1]$. Since one output neuron describes the secondary task *reading* and the other neuron describes the task *video*, a simple comparison is finally used after the ANN to determine the estimated secondary task.

[6] The documentation can be found on https://de.mathworks.com/help/nnet/index.html.

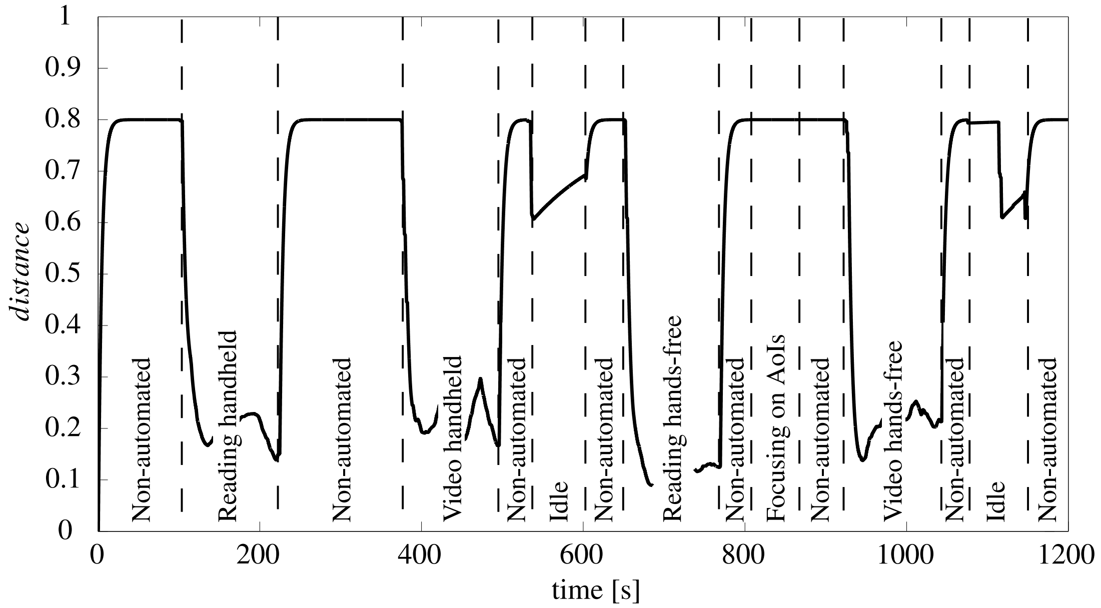
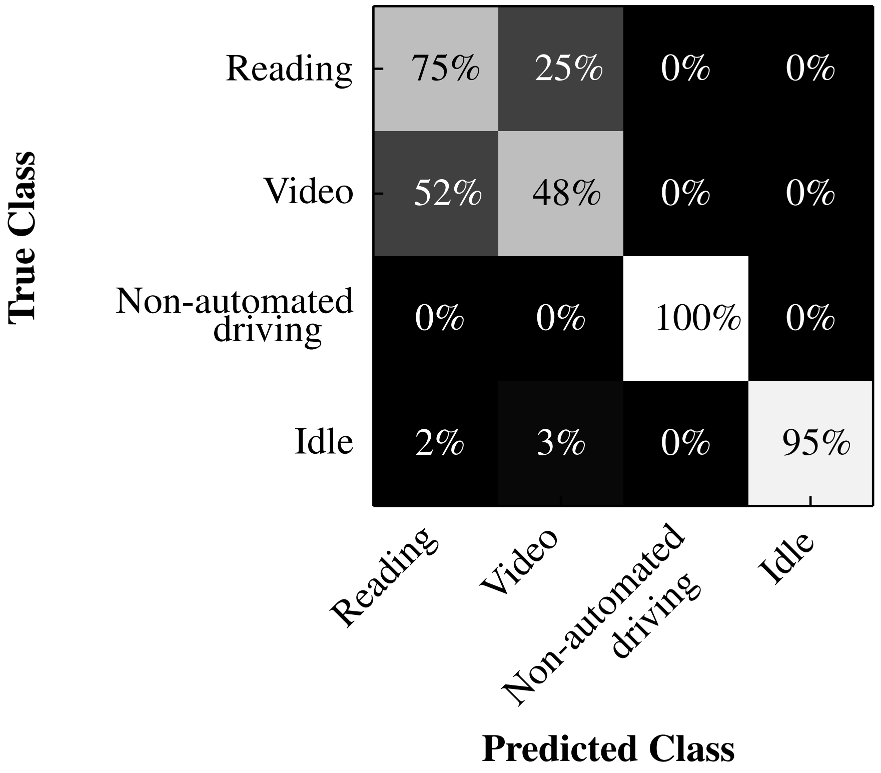


Figure 5.23: The distance signal between the online updated hash table and the a priori learned reference table is plotted over a complete test drive. The dashed lines visualize the shift between the different tasks during the test drive labelled with the corresponding task name.

5.5.4 Evaluation

Following the modification of the scanpath-based approach in the previous Subsections 5.5.1, 5.5.2, and 5.5.3 an evaluation of the online variant is performed based on the data of the real-driving study presented in Subsection 2.4.5. In this study, the subjects performed secondary tasks including reading, watching a video, and being idle while driving in a conditionally automated setting. Moreover, the tasks *reading* and *video* were performed on a handheld and hands-free device. In addition, there were route sections forcing the driver to take-over the control of the vehicle. The start and end points of each task were manually labelled by an instructor sitting on the backseat of the vehicle. These labelled areas are used as ground truth data for the verification. As for the application of the ANN, the number of hidden layers was optimized over the data set and resulted in the value of 15 hidden layers. For the following results, the sequence size was set to $m_{seq} = 15$ s.

Figure 5.24 shows the confusion matrix of the classification of the secondary tasks. For the sake of completeness, the recognition of manual driving phases is included. As discussed before, the separation in manual and automated driving phases is based on a binary signal provided by the automated driving function and, therefore, provides a perfect classification accuracy. Moreover, the class *idle* is also nearly at its optimum with a true positive rate of 95%. This high true positive rate is achieved by combining the online cluster detection method of Section 4.5 and the classification based on the Euclidean distance between the current hash table of the subject and the reference hash table learned a priori. Each time



	Reading	Video	Non-automated driving	Idle
Reading	75%	25%	0%	0%
Video	52%	48%	0%	0%
Non-automated driving	0%	0%	100%	0%
Idle	2%	3%	0%	95%

Figure 5.24: A 4 × 4 confusion matrix based on the classification of the online scanpath-based DAR. The approach was applied on the complete data set of the real driving study.

a driver is detected as not focusing on the AoI of the handheld or hands-free device, the model deduces that the driver is not performing any secondary task. In addition, if the driver is focusing on the handheld or hands-free AoI, the Euclidean distance is analyzed. If the distance value exceeds the specified limit of 0.4, an idle driver is classified. This approach falsely recognizes a driver engaged in one of the secondary tasks for only 5% of the samples. However, the most challenging part of the DAR is shown in the upper left corner of the confusion matrix. While 75% of the reading scenarios are correctly recognized, the classification accuracy of the *video* task is down to guessing probability.

Following this first evaluation, the scanpaths for both tasks *read* and *video* were compared in greater detail. It was found that there are significant differences between drivers performing many EoR gazes and drivers performing none or just a few EoR gazes. Figure 5.25 shows an example for each type of driver with regard to the frequency of EoR gazes. As can be seen on the upper plot, the scanpath shows the typical sawtooth pattern for reading sequences. On the plot below, another scanpath of a reading sequence is plotted. Here, the driver performed many EoR gazes, highlighted in red. These EoR gazes act as noise disturbing the pattern recognition. That means that the more EoR gazes are performed, the less visible the patterns and the less data is available for the DAR. In addition, there is another effect reducing the classification performance, in particular of the *video* task.

Although data recorded during EoR gazes is discarded and not applied for the classification of the secondary task, the EoR gazes influence the scanpath patterns significantly. More precisely, the recorded samples during the gaze shift of each EoR gaze describe a scanpath similar to the sawtooth pattern when reading a text. These gaze shifts before and after each EoR gaze are not counted to the EoR gaze and, therefore, are applied to the DAR. As a consequence, the more EoR gazes are performed, the more sawtooth similar scanpaths appear during the video task, reducing the classification accuracy between the tasks *read* and *video*. To prove this assumption, the EoR gazes are discarded along with 10 frames prior to and after each EoR gaze and the confusion matrix is recalculated in Figure 5.26. As it can be seen, the true positive rate for both tasks is increased and at about 75% is now on a similar level as the classification performance of the offline variant in Subsection 5.4.2.

As mentioned at the beginning of this subsection, the sequence size was set to $m_{seq} = 15\,\mathrm{s}$. In general this means that the model needs 15 s of recorded data before a robust classification can be performed. In Figure 5.27, the sequence size m_{seq} was decreased and the true positive rate of each secondary task as well as the common accuracy is plotted. As expected, the accuracy and true positive rate decreases with the smaller sequences. For a sequence size of 10 s, the overall accuracy decreases only by about 2% to 73% indicating the applicability of even smaller sequences.

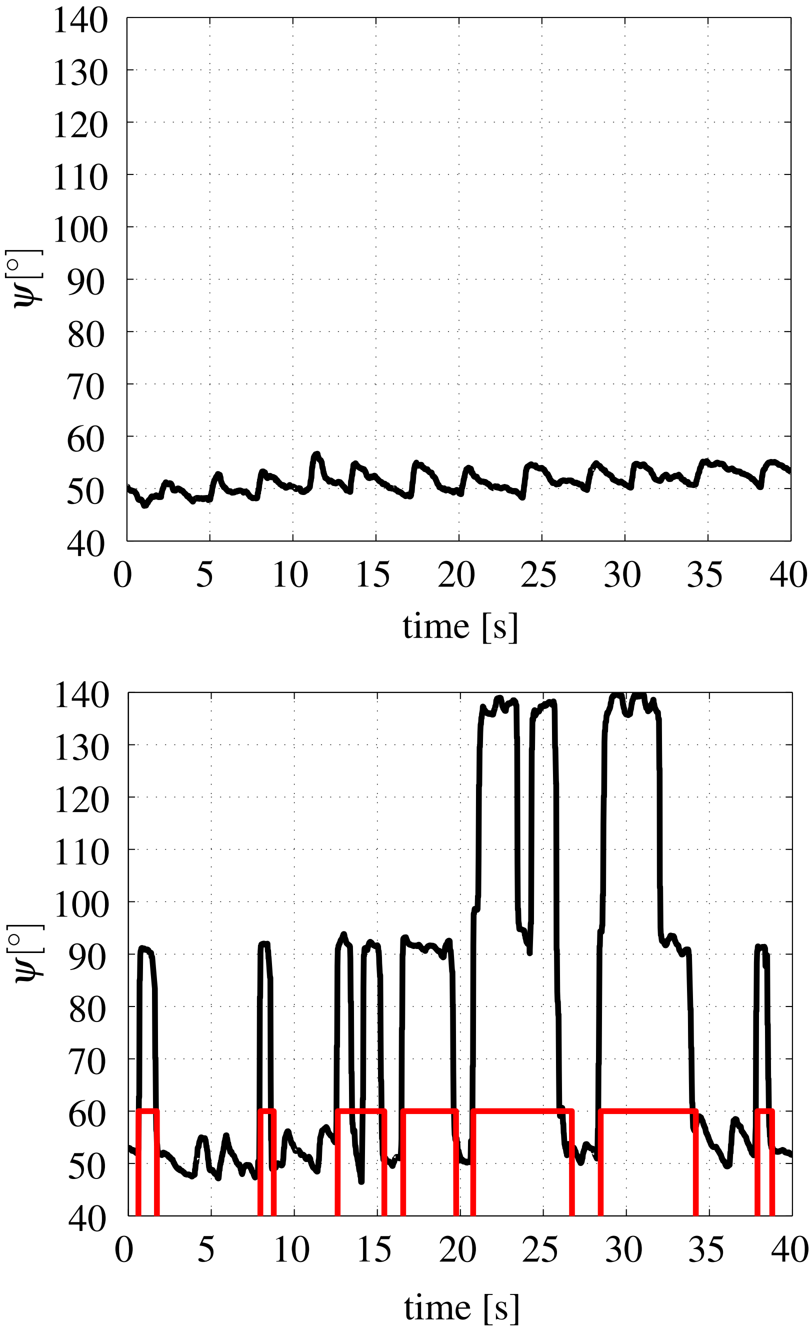
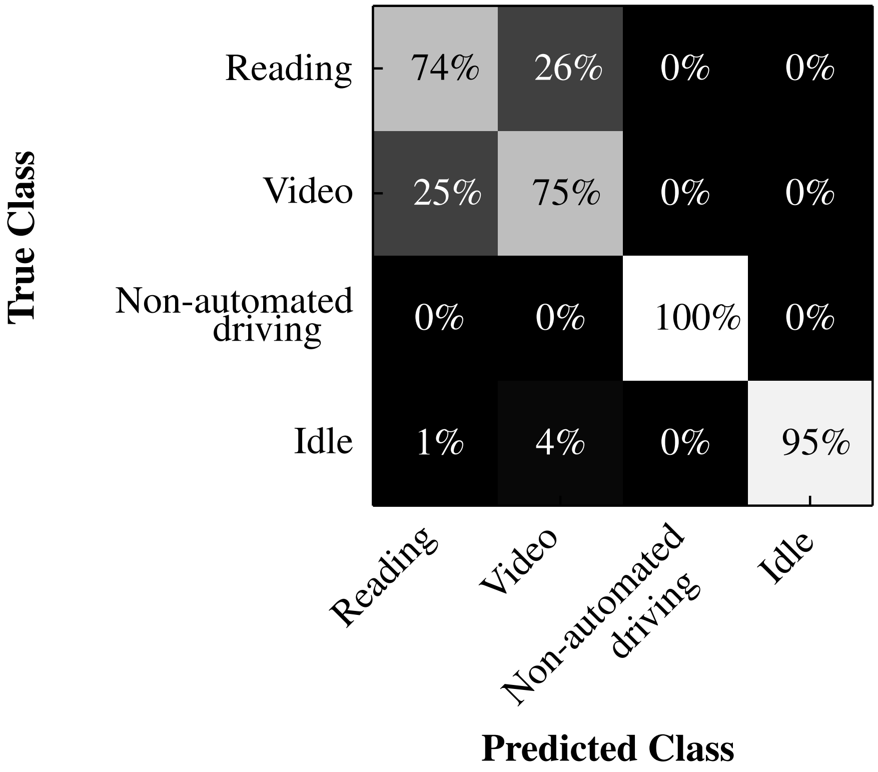


Figure 5.25: Two plots of a reading sequence of two different subjects. The subject recorded on the upper plot performs no EoR gaze resulting in the distinct reading pattern. The subject recorded on the lower plot performs many EoR gazes (highlighted with red) which exacerbates the DAR.

However, an additional reduction of the 5 s sequence leads to a significant misbalance between the two classes. While the true positive rate of the reading class increases to about

85%, the video class is reduced to 52%. This is usually a sign for a higher weighting of the reading class by the ANN assigning most of the samples to the class with the higher weighting.



True Class \ Predicted Class	Reading	Video	Non-automated driving	Idle
Reading	74%	26%	0%	0%
Video	25%	75%	0%	0%
Non-automated driving	0%	0%	100%	0%
Idle	1%	4%	0%	95%

Figure 5.26: The same 4x4 confusion matrix as in Figure 5.24 after the elimination of the EoR gazes.

In summary, the scanpath-based DAR was successfully adapted to an online application in the vehicle reaching a similar classification performance as the online variant. The EoR gazes, especially the corresponding gaze shifts, have to be taken into account for an adequate result. Finally, it should be noted that the classification performance for sequences shorter than 10 s do not appear to contain enough relevant information to separate detailed secondary tasks robustly. This result is in alignment with the evaluations of Section 5.4.

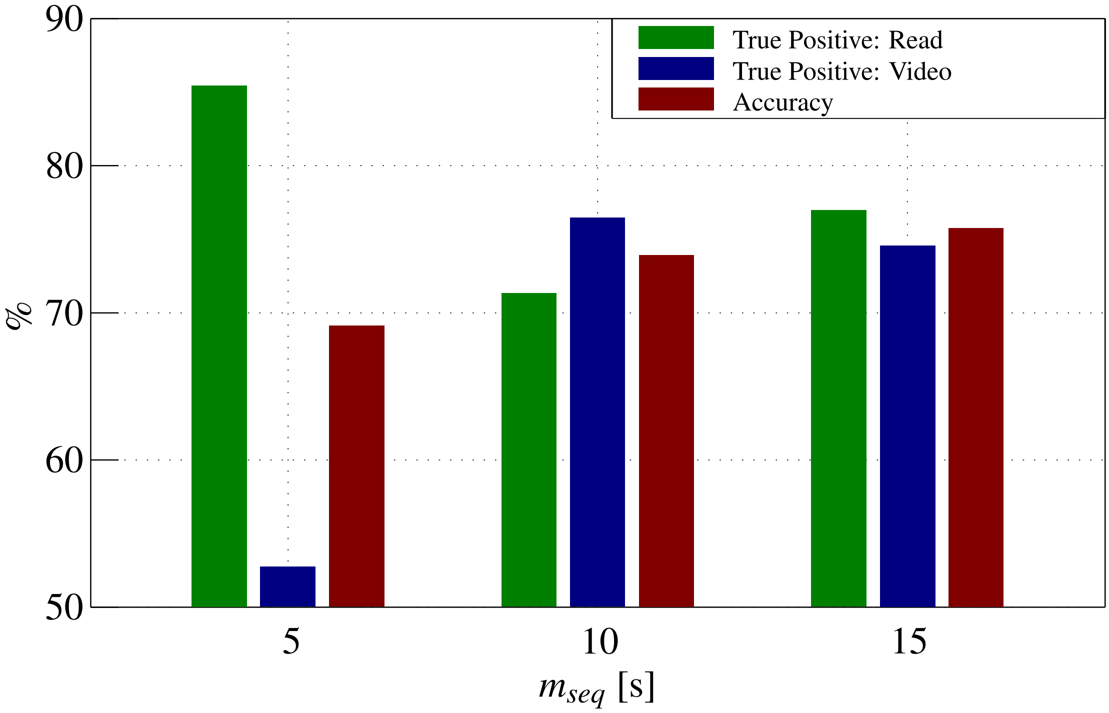


Figure 5.27: Bar diagram of the true positive rate of the classes *read* and *video* and their common accuracy. The statics are calculated for 5 s to 15 s sequences to determine the smallest sequence with reasonable performance.

5.6 Summary

In this chapter, methods for driver-activity recognition were described. After summarizing the current state-of-the-art, two approaches were proposed for further investigation. The first approach is based on chronologically independent eye and head features derived from saccades, fixations, eye blinks, and the head pose. Further, features especially designed for conditionally automated driving scenarios were introduced which are based on occurring gaze behaviors such as the EoR gazes. These eye features directly linked with the mentioned gaze behavior and two head features reflecting the head direction of the driver were of particular benefit and enabled a significant improvement of the classification of secondary task such as reading or watching a video. The second approach is based on the driver's scanpath and therefore incorporates the chronological order of the eye movements. The scanpath-based approach uses SAX-patterns to highlight specified gaze patterns before extracting small subsets of the scanpath called words. Finally, the frequencies of these words are compared in a classification step. A comparative evaluation of both approaches revealed the superior performance of the scanpath-based approach even for online scenarios. Due to the abstracted features incorporating the temporal order of the gaze patterns for classification, the scanpath-based approach increased the accuracy of the classification performance by about 19% compared to the first approach in online settings. As a consequence, the scanpath-based approach was modified for the application in a testing vehicle and evaluated with data from a real driving study. It was shown that the approach classifies the secondary task robustly even in the light of realistic online scenarios.

Future work with both approaches involves the detection of additional secondary tasks including the interaction with other passengers. Moreover, both approaches could further benefit from including other sensing modalities, e.g. hand gestures and manual interactions with integrated devices [32], to enable a more accurate and robust classification even in case of additional tasks. Further improvements of the classifiers by means of more data sets could be enabled by randomized features [33] while keeping the computational costs tolerable. Especially for the scanpath-based approach, different SAX patterns should be investigated which may outperform the vertical SAX pattern of this study.

The scanpath-based method for driver-activity recognition will be used to extract features for a classification of the driver's take-over readiness in Chapter 6.

6 An Automated Classification of Take-Over Readiness

In the previous chapters, approaches for recognizing the performed secondary task and methods for detecting EoR gazes were introduced and investigated for application in conditionally automated vehicles. These previous chapters can be seen as preparation for the following analysis of an approach which classifies the take-over readiness of the driver during conditionally automated driving scenarios based on features provided by these introduced approaches. The goal of this chapter is to create a first prototype of an ADAS able to estimate the take-over readiness of the driver. However, training a classifier capable of differentiating between high- and low-quality take-over situations with adequate performance requires features derived from not only driver monitoring but also from the present traffic situation. Moreover, formal specification of what is meant by high and low quality regarding take-over situations is required.

To provide an overview of the relevant impact factors regarding take-over readiness and to outline the lag of appropriate automated methods to estimate the driver's take-over readiness, Section 6.1 summarizes the state-of-the-art concerning these topics. Section 6.2 introduces the architecture of the planned ADAS and describes how to obtain the missing aspects: the measures for assessing the take-over quality in Subsection 6.2.2 and the features based on the complexity of the current traffic situation in Subsection 6.2.1. Finally, the choice of the classifier as well as the evaluation of the classifier and the proposed novel ADAS are discussed in Section 6.3. The results of this chapter are based mainly on the author's journal publication [18].

6.1 Take-Over Readiness - Influences and Automated Detection Approaches

6.1.1 What influences the Take-Over Readiness of a Driver?

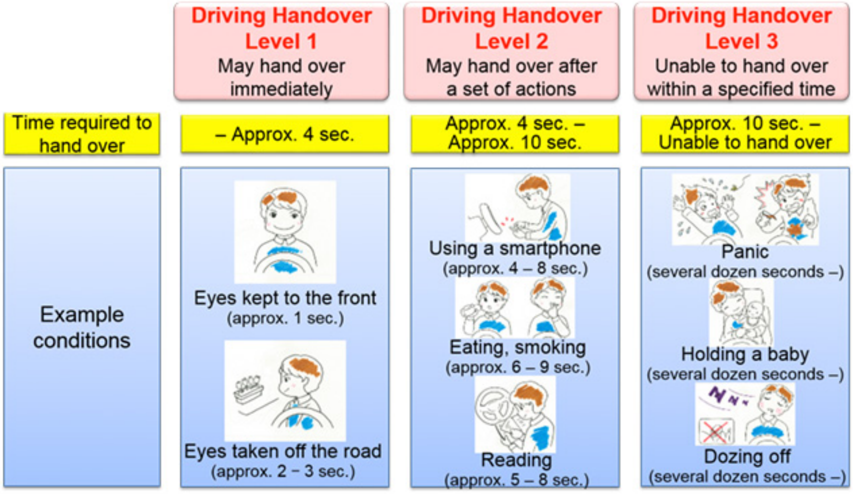
Many recent studies have investigated the most significant factors influencing the driver's take-over readiness during conditionally automated driving scenarios. In [8], Radlmayr et al. conducted a driving simulator study to evaluate the impact of the traffic situation and non-driving related tasks on take-over situations. Their results showed that take-over quality and traffic situation difficulty correlate. The number of collisions increased significantly for situations with prevalent high traffic density, and was boosted further by the performance of non-driving related tasks. Note that the performed tasks were of artificial nature, e.g. the visual surrogate reference task [119].

More realistic secondary tasks were performed in another driving simulator study by Zeeb et al. [9]. The authors examined the impact of various secondary tasks on the take-over quality, including tasks such as writing an email, reading news, watching a video, and listening to music. The reported data implied that watching a video or reading a news article deteriorates the take-over quality while response times (such as the time to get the hands back on the wheel) show little to no impact. The authors stated that motor processes are performed reflexively and that the take-over quality is primarily influenced by the driver's cognitive comprehension of the situation. In a follow-up study [97], Zeeb et al. focused on secondary tasks with a varying level of manual or cognitive workload. They reported that engagement with a handheld device significantly delayed the response times of the driver and deteriorated the overall take-over quality. Variations in the cognitive task load of the non-driving related tasks showed a similar effect but not for each type of take-over scenario.

Besides the influence of secondary tasks, Zeeb et al. [11] also analyzed the impact of Eyes-on-Road gazes on the number of collisions in take-over situations. More specifically, the authors identified three groups of drivers based on the parameters frequency and duration of the glances: low-, medium-, and high-risk. The authors found a significant difference between the identified three groups in terms of number of collisions in a demanding take-over scenario [11]. The drivers in the high-risk group noticed the simulated crash in their lane and the necessity to act immediately. However, since their gaze was neither on the road nor on the surrounding traffic environment, these drivers overlooked the oncoming vehicles on the adjacent left lane and collided. In contrast, drivers of the group low- or medium-risk performed the correct braking maneuver far more often. In a study by Feldhütter et al. [120], the authors identified the automated drive duration as an influencing factor on gaze behavior. For uninterrupted durations of more than 20 minutes, the driver began to let their gaze wander to compensate for the monotony. However, neither the gaze behavior nor the actual duration had an impact on the take-over quality in that study.

6.1.2 Automated Detection of the Take-Over Readiness

There is hardly any literature available describing systems for automatically classifying the take-over readiness of a human operator working with automated systems. In general, to compensate for the shortcomings in take-over situations, the literature investigated primary design concepts for the human-machine interfaces [121]. Clear instructions and optical and visual support should ensure that the correct mental model is provided. In a work by Gold et al. [122] the authors analyzed the benefit of gradual automated driving function degradation if the take-over situation permits it. This approach further facilitates the take-over by reducing the driver's number of tasks. The only known method of the literature covering the topic of an automated classification of the take-over readiness was introduced by Nilsson et al. [123]. The authors suggested defining a safe transition from an automated to a manual driving level based on the driver controllability set (DCS). The DCS is defined as a subset of the vehicle's state space and is individually learned during manual driving phases. Simply put, it represents the individual driver's capabilities. As long as the vehicle state is within the DCS during transitions, the driver's skills are sufficient to perform a safe



	Driving Handover Level 1 May hand over immediately	Driving Handover Level 2 May hand over after a set of actions	Driving Handover Level 3 Unable to hand over within a specified time
Time required to hand over	– Approx. 4 sec.	Approx. 4 sec. – Approx. 10 sec.	Approx. 10 sec. – Unable to hand over
Example conditions	Eyes kept to the front (approx. 1 sec.) Eyes taken off the road (approx. 2 – 3 sec.)	Using a smartphone (approx. 4 – 8 sec.) Eating, smoking (approx. 6 – 9 sec.) Reading (approx. 5 – 8 sec.)	Panic (several dozen seconds –) Holding a baby (several dozen seconds –) Dozing off (several dozen seconds –)

Figure 6.1: The figure taken from [124] shows the three handover levels of the Omron's driver concentration sensing technology with the corresponding take-over time and example conditions.

take-over. However, a safe transition cannot be guaranteed if the vehicle state leaves the DCS. The proposed approach was evaluated based on real driving data during activated adaptive cruise control (ACC). Since the boundaries of the DCS are updated online during "normal" manual driving scenarios, the system is constructed conservatively. That means that there may be many false alarms for situations where the driver could still be in control. In addition, it is questionable if the driver's capability during manual driving is without limitations compared to the driver's capability during take-over situations. For example, drivers usually keep their eyes on the road while driving manually, which is not a necessary requirement in conditionally automated driving scenarios. To which extent this reduced situation awareness influences the driver's control in similar situations is still unknown and not covered by the proposed model in [123]. A similar approach to the one proposed in this work was introduced by OMRON Corporation in 2016 and described as *driver concentration sensing technology* [124]. OMRON developed an onboard sensor to monitor the driver's various motions and conditions. Depending on the detected conditions the system classifies whether the driver is able to take over safely in an approximated, coarse time interval (cfg. Figure 6.1). However, the shown test cases and evaluations of this prototype were based solely on artificial situations, i.e. the test subjects were sitting in a lab environment or fixed driving simulator and pretended to fall asleep, to use a smartphone, to drop objects, etc. Their gestures were performed in an exaggerated manner. Further, there was no individual separation within the various use-cases, e.g., a take-over time of approximately 4 s was assumed for each driver taking their eyes off the road, independent of the preceding driving behavior. No information was provided on whether the two use-cases *Using a smartphone* and *Reading* were even distinguishable.

6.2 Prototype of a novel ADAS

The overall architecture of the ADAS proposed in this work is shown in Figure 6.2. The goal of this approach is to control the driver's attention to the road and to enable additional secondary tasks as long as the driver follows the instructions of the ADAS. All features are continuously extracted from three sources: the secondary task, EoR gazes, and the traffic situation. The driver's responsiveness is classified based on these features for each time step. Moreover, the architecture is modular, i.e. the applied eye- and head-tracking systems and the methods for calculating the features are interchangeable if the same features and signals can be provided. The classifier itself has to be trained in advance to account for aspects such as individual gaze and take-over behavior using a preferably widespread training set of take-over situations. The prototype is configured so that for a classified high take-over readiness no further measures are required since it is assumed that the driver is aware of the current traffic situation and able to take over adequately. As a consequence, the driver is not disturbed by any warning and the system will still enable the driver to perform the whole set of secondary tasks. However, if a low take-over readiness is classified the driver will be asked to perform gazes towards the road for reorientation. If the driver ignores this warning, the set of possible secondary tasks could be reduced to the less demanding ones.

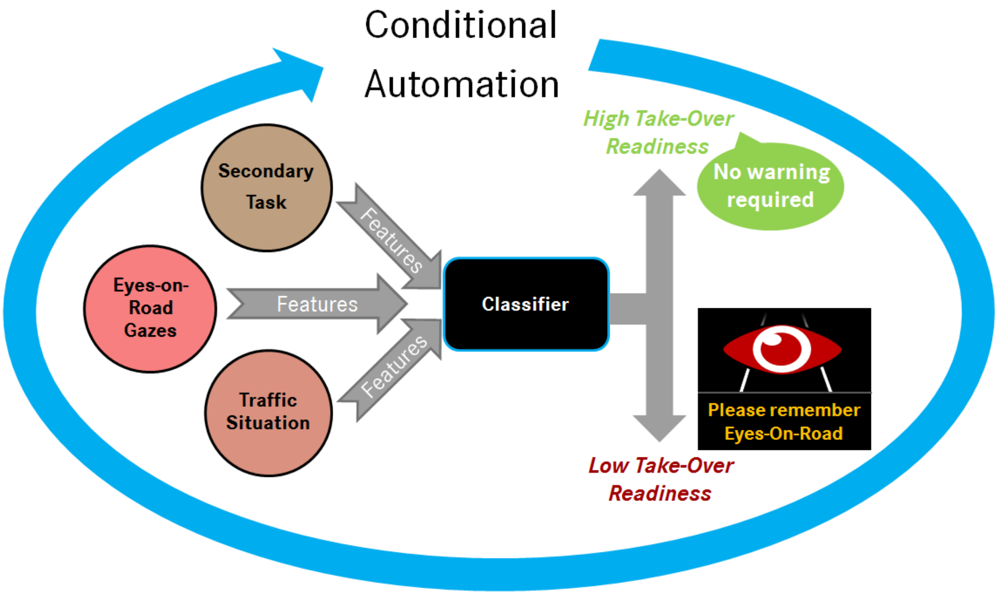


Figure 6.2: Overall architecture of the proposed ADAS prototype.

In total, four features are extracted from both information sources "secondary task" and "EoR gazes" describing the current activity and situation awareness of the driver. These features derived from secondary tasks and EoR gazes can be extracted by means of the methods introduced in Chapter 4 and Chapter 5. Table 6.1 summarizes the applied features including a short explanation and the possible values. As it can be seen, there is one

Feature	Description	Values
Situation Complexity	Level of complexity of the traffic situation	$0 = easy$ $1 = medium$ $2 = high$
Last Gaze	Time since last EoR gaze	Time in seconds
Number Gazes	Number of EoR gazes	$\in \mathbb{N}$
2nd Task	Task performed by the driver	$0 = idle$ $1 = video$ $2 = read$
Manual Demand	Type of manual demand	$0 = Handheld$ $1 = Handsfree$

Table 6.1: Extracted features for the classification of the driver's take-over readiness.

feature which is not based on driver monitoring. Instead, feature ”Situation Complexity” is derived from the traffic situation and describes the current level of complexity of the traffic situation. In the following subsection, the generation of this feature is explained in detail.

6.2.1 Determining the Complexity of a Traffic Situation

An important aspect for determining the take-over readiness is the complexity of the present traffic situation. More specifically, the following question needs to be answered: How complex would be a take-over situation occurring in the very moment? If the situation is not demanding at all, e.g., a take-over situation on a straight highway without any traffic, no critical situation will occur even in the case of a slow-acting or an idle driver. In such scenarios, the automated driving function would slowly bring the vehicle to a standstill. On the other hand, such a driver behavior may lead to critical situations in case of more complex situations, e.g., another vehicle cuts in on the ego-vehicle right after the take-over time. Thus, depending on the actual traffic situation during a take-over, the behavior of the driver has to be judged differently. This impact of the current traffic situation on a take-over situation was investigated and proofed by Radlmayr et al. [8] in a conditionally automated driving simulator study with four take-over situations. Especially a high traffic density showed a severe negative impact on the take-over performance. Therefore, the complexity of the current traffic situation will be considered for the proposed classification of the take-over readiness as additional feature for the classifier.

There are various aspects influencing traffic situations which exacerbates a direct comparison of these situations. Hence, many studies focus on separating traffic situations by means of classification schemes. Such schemes are suitable tools for categorizing the complexity of the analyzed traffic situation. One of the first thorough investigations concerning this topic was done by von Benda et al. [125]. In this work, a classification scheme by means of a multi-dimensional scaling was proposed. It was based on the situation assessment of the

driver, i.e. which aspects are considered to be relevant by the driver and which attributes are the most salient ones. For the classification scheme, an empirically established level of hazard for every traffic situation as well as further situation conditions, such as the traffic density, the horizontal and vertical course of the road, the weather conditions, etc. were used to represent the various dimensions of the approach. Hence, a structural representation and separation of the respective traffic situation was possible. However, due to the multiplicative relation between the dimensions, more than three million different traffic situations can be described which reduces the comparability and generalizability.
To enable the application of the above described classification scheme from [125] in the field, Fastenmeier et al. [126] reduced the dimensionality by discarding some of the considered aspects. Furthermore, the authors focused on assigning a corresponding workload to the situations by means of the questionnaire proposed in [127], grouping traffic situations by complexity: low, medium, or high. The result of this work would be best described as formulary. Traffic situations, described by aspects such as weather and visibility conditions, number of lanes, type of the road (highway, urban, rural) or the speed limit, are listed with corresponding workload indices. Based on this index, the traffic situation can be categorized in one of the three mentioned complexity groups. This classification scheme, introduced by von Benda and simplified by Fastenmeier et al. seems to be a convenient approach to assign suitable levels of complexity to the traffic situations occurring in our study.

For this work, the scheme has to be adapted to the conditionally automated driving simulator study. As a consequence, many of the aspects can be ignored, such as pedestrian crossings or junctions since the study was performed on a high-way. Moreover, some aspects could be assumed to be constant such as the number of lanes. In Table 6.2, the remaining aspects of the scheme are summarized and aligned with their possible values. In total, four aspects of the classification scheme are considered: the *curvature of the road*, the *traffic density*, *special weather conditions*, and *perils*. Fastenmeier et al. [126] distinguished between *straight*, *sharp left/right* and *wide left/right* curves as possible states with regard to the aspect *curvature of the road*. The original aspect *traffic density* could assume the four states *low*, *normal*, *dense*, and *traffic jam*. The state *traffic jam* was deleted since it is irrelevant for our study. The aspect *special weather conditions* usually summarizes particular visibility conditions, such as rainfall at night. However, in our study this aspect only describes the presence of strong crosswind. The last aspect *perils* is applied in case of the occurring braking maneuver of a leading vehicle during one of the take-over situations. Following the taxonomy of the situation complexity in [126], traffic situations on a straight or curvy highway are considered as not critical and easy to handle for the driver. Similar conclusions can be drawn for situations with a high traffic density. Although Fastenmeier et al. point out that increasing traffic density has a systematic negative impact on all traffic situations, it is not a sufficient condition for labeling the corresponding situation as complex. Hence, the occurrence of each of these criteria by its own will indicate a traffic situation with a low complexity. However, if both criteria are present at the same time, the complexity is increased to a medium-high level. For the whole course of the driving simulator study, only two traffic situations are classified as highly complex, namely the two last

Aspect	States
Curvature of the Road	0 = straight 1 = wide curve 2 = sharp curve
Traffic Density	0 = low 1 = normal 2 = dense
Special Weather Conditions	0 = no special conditions 1 = crosswind
Perils	0 = no peril 1 = occuring peril

Table 6.2: The table summarizes the remaining aspects of the original study by Fastenmeier et al. [126] applied in this work. The right column describes the corresponding set of possible states of each aspect.

take-over situations *Cross-Wind* and *Braking* described in Subsection 2.4.4. In accordance with the approach in [126], the complexities of these take-over situations were determined via the questionnaire proposed in [127] and completed by three raters. For the take-over situation *Cross-Wind*, the vehicle is passing through a sharp left curve on the rightmost lane while there is upcoming traffic on the adjacent left lane. At the same time, the vehicle is drifting to the right breakdown lane due to a strong crosswind from the left. Especially the aspect of the special weather condition increases the challenge concerning the vehicle guidance to keep the vehicle in the lane to prevent any collision. The second take-over situation *Braking* is characterized by a high traffic density so that no lane change to the left is possible. Furthermore, at the beginning of the take-over situation a vehicle is going into the rightmost lane in front of the subject's vehicle and decelerates significantly. Compared to the take-over situation with cross-wind, where the vehicle guidance by the driver was the challenging part, this situation is particular demanding in terms of the visual perception (noticing as early as possible the strong braking vehicle) and decision making processes of the driver (braking since the lane is blocked). These two aspects, namely the demanding visual perception and the decision making processes, are typical signs for highly complex traffic situations according to Fastenmeier et al. [126].

Since the actual ground truth of any simulated aspect is known in a driving simulator study, an automated realization of the classification scheme is easily implemented as a lookup table. The feature of the situational complexity with the possible states low, middle, and high was calculated for the complete route. Since the subjects drove solely on a highway without any ramps or intersections, the complexity was classified as low for most of the time. The complexity level increased to medium only in sharper curves during high traffic density. Situations with a high complexity occurred only for the latter two take-over situations due to the strong cross-wind and the hazardous maneuver of the other vehicle cutting in. The situational complexity represents the last feature with regard to the classification of the driver's take-over readiness.

6.2.2 Measures of the Take-Over Quality

By means of the methods for EoR detection in Chapter 4, DAR in Chapter 5, and the approach described in the previous subsection, all features required for the classification of the driver's take-over readiness can be extracted of the data of the KoHAF study. These features in combination with the two class labels *high take-over quality* and *low take-over quality* enable the training of the classifier. Of course, data of both well-performed and inadequate interventions in take-over situations are required to train and evaluate the classifier. However, high or low take-over quality must be formally defined. For this purpose, objective driving parameters were selected to provide a measure for the take-over quality. Depending on the take-over situation, different parameters need to be considered. For example, for the *Cross-wind* take-over situation relevant parameters would consider the lateral steering behavior of the driving whereas for the *Braking* situation parameters concerning the longitudinal behavior of the vehicle would be appropriate. Since there are already suitable parameters discussed in literature, this study follows the parameter proposal of Zeeb et al. [97]. In addition to these parameters, the parameter *Performed lane change* for the situation *Straight* is included. The take-over situations and corresponding parameters are listed in Table 6.3. All parameters of the situation *Cross-wind* were calculated for a seven seconds interval after the take-over request during which the cross-wind occurred. For the situation *Braking*, all parameters are analyzed during a six seconds interval after the take-over request during which the leading vehicle decelerated.
To interpret the parameters with regard to the quality of a take-over maneuver, there must be a defined range specifying a high or low take-over quality for each parameter. The control group, whose members did not perform any secondary tasks and instead observed the traffic environment, was used for this purpose. The take-over quality was validated by three raters for all subjects of the control group to ensure the usage of only high-quality take-over interventions. The ratings indicated that all but one subject who ignored the instructions performed a high-quality take-over. For all but one parameter, the mean μ_c and standard deviation σ_c of all the subjects in the control group were calculated (see Table 6.3). Afterwards, the upper and/or the lower threshold to the corresponding driving parameter was calculated over $\mu_c \pm 2\sigma_c$. Although minor deviations against normal distribution were observed for some of the parameters, a normal distribution can be assumed for each parameter. As a consequence, 95.45% of the data of the control group are contained in the range $\mu_c \pm 2\sigma_c$. If the value of a parameter lies within this calculated range, the parameter indicates a high take-over quality. If the value exceeds the calculated range, a low take-over quality is considered for this parameter. Hence, a normalization of each parameter is performed to enable the comparison between the three situations. For parameter *Performed lane change* of situation *Straight*, a lane change was always considered as low-quality take-over. However, one parameter is not significant by itself. For example, some drivers started to brake right after the take-over request resulting in a small time to first braking which usually indicates a high-quality take-over. However, some of these drivers braked not enough so that the distance to the leading vehicle became quite small indicating

a low-quality take-over. Hence, a low or high take-over quality is only assigned to the take-over situation by means of a majority decision. In case of the situations *Braking* and *Cross-wind*, at least two of the three parameters have to indicate low or high take-over for a low or high take-over quality to be assigned to the overall situation. In case of the situation *Straight*, at least one of the two parameters has to be outside the range for the take-over situation to be labelled as low take-over quality.

Situation	Driving Parameter	Control Group		Experimental Group		Both
		μ_c	σ_c	μ_e	σ_e	$\in \mu_c \pm 2\sigma_c$
Straight	Performed lane change [yes/no]	-	-	-	-	99% (80)
	Maximum deviation from lane center [m]	0.18	0.1	0.24	0.15	96% (78)
Cross-wind	Time to first steering [s]	1.96	0.28	2.53	0.56	60% (49)
	Minimum time to lane crossing [s]	1.86	0.56	1.35	0.59	85% (69)
	Maximum deviation from lane center [m]	0.76	0.12	1.17	0.43	49% (40)
Braking	Time to first braking [s]	2.86	0.38	3.09	0.53	88% (71)
	Minimum distance to leading vehicle [m]	12.1	3.66	9.16	4.8	78% (63)
	Minimum time gap to leading vehicle [s]	0.47	0.12	0.37	0.17	73% (59)

Table 6.3: Take-over situations and corresponding driving parameters to determine take-over quality. The mean and standard deviation is given for the control (μ_c, σ_c) and experimental group (μ_e, σ_e). The last column shows the percentage and absolute number of situations of both groups in which the corresponding parameter is included in the defined high-quality range.

6.3 Evaluation of the Advanced Driver Assistance System

The last step for creating the ADAS shown in Figure 6.2 is the choice and training of the classifier discussed in Subsection 6.3.1. It is crucial to outline the challenges for this classifier which significantly impact the choice of the approach. Moreover, the classification performance of the introduced features has to be analyzed in detail to show their potential for classifying the driver's take-over readiness. However, Subsection 6.3.2 shows that not only the applied features but also the type of the intervention during the take-over situation influence the classification. The final Subsection 6.3.3 of this evaluation focus on the prototype of the ADAS described in Section 6.2.
For the following evaluations of the classifier and its characteristics, Ground Truth data was applied in Section 6.3.1 and Section 6.3.2. However, features derived from the measurement data of the simulator study were applied to test the prototype of the ADAS in Section 6.3.3. The following results were calculated by means of a Leave-N-out cross validation with $N = 5$ to increase the number of combinations compared to a validation with $N = 1$. Hence, five subjects were removed from the training set and used for the evaluation. Each of the 81 subjects[1] including the control group with 14 subjects experienced three take-over situations, resulting in a total of 243 situations. The number of iterations was limited to 5000 since the evaluation over all combinations of $\binom{81}{5}$ is extremely time-consuming and the result converges already for smaller numbers. Following the approach in Section 6.2.2, 60 take-over situations were assigned a low take-over quality whereas 183 situations were assigned a high take-over quality. In detail, for the simple take-over situation *Straight*, only one subject showed a low-quality take-over. For the situation *Cross-wind*, 37 subjects performed a low-quality take-over whereas for the situation *Braking* 22 drivers showed an inappropriate intervention. Moreover, during 25 situations performed by 19 different drivers of the experimental group, a gaze at the road was performed shortly before the take-over occurred. To prevent a higher weighting of the class representing situations with high take-over quality, random high-quality take-over situations were discarded to obtain the number of data sets similar to that of low take-over quality situations. The recording of realistic take-over situations is an expensive task in terms of time and money, and balancing further reduces the already low amount of training data. Thus, the applied training set is balanced and subject-independent, but contains relatively few take-over situations.

6.3.1 Analysis of Classifiers and Features

One possible method to deal with this challenging machine learning task is to select an appropriate classifier and a small number of significant features. Various classification methods are applied to the task of determining the take-over readiness: SVM with a linear and RBF kernel with regularization factor $C = 1$, linear discriminant, Naive Bayes, and the k-nearest neighbors algorithm (KNN). These algorithms were selected for their reported adequate classification performance with limited training data in previous studies [128]. Only the five features described in Section 6.2 were applied as input. In addition, the combination of these classifiers with a preceding feature selection by means

[1] 45 males/36 females, mean age of 38 years (range 20-58, SD=11)

Method	FCBF	Accuracy	F1 score	∅ Time
KNN	✗	0.70	0.69	53 ms
RBF SVM	✓	0.76	0.75	5 ms
Linear SVM	✓	0.79	0.77	4.6 ms
Naive Bayes	✗	0.68	0.75	4.4 ms
Linear Discriminant	✗	0.74	0.72	1 ms

Table 6.4: Performance of various classifiers in determining the take-over readiness for the Leave-5-out cross validation.

of the FCBF was investigated to further reduce the number of features. The evaluation is based on the Ground Truth data obtained by video; the EoR gazes and secondary tasks for the 60 s-interval before each take-over are labelled by three raters. Since the feature of the situational complexity is based solely on the simulated data, the automated realization of the classification scheme described in Subsection 6.2.1 always provides Ground Truth data. Table 6.4 shows the accuracy and F1 score of these methods and highlights the highest measures. A checkmark indicates that the result was improved by applying the FCBF. The averaged computation time for one iteration of the Leave-5-out cross validation was calculated on an Intel Xeon(R) with 2.4 GHz. According to Table 6.4, a preceding feature selection improves only the SVM performance. These results confirm the evaluation of Forman and Cohen [128] indicating an excellent classification performance of the linear SVM. However, the performance of the SVM with RBF kernel and the linear discriminant is decreased only slightly by about 3% and about 5% for the accuracy and F1 score, respectively, compared to the linear SVM. Furthermore, if the computation time is considered as a relevant design criterion for the ADAS, the linear discriminant has to be taken into account, since its execution time is about five times shorter than for SVM approaches. The SVM with a linear kernel was applied to obtain all the following results.

In Figure 6.3, the accuracy and the F1 score of the take-over classification for all feature combinations based on the Ground Truth data are shown. The first three combinations only contain one type of features. As it can be seen, an accuracy of 71% and an F1 score of 0.6 can be reached using only the situation complexity. This provides a slightly more accurate classification than the DAR features. However, the features of DAR increase the F1 score by 0.06. Compared to these two combinations, EoR features show a significantly lower performance for both measures and indicate to be not appropriable as single features. The fourth combination includes the features of the EoR gazes and the DAR and shows a significantly increased F1 score of 78% while the accuracy decreases slightly by 1%. For all of the following combinations, the curve of the accuracy increases steadily to 79% whereas the F1 score remains almost stable for the third and fourth combination. Figure 6.3 indicates that a robust prediction of the driver's take-over readiness can be based solely on neither the current traffic situation nor on driver-related features. It is shown that using only the driver-independent feature of the situation complexity, the F1 score has an extremely

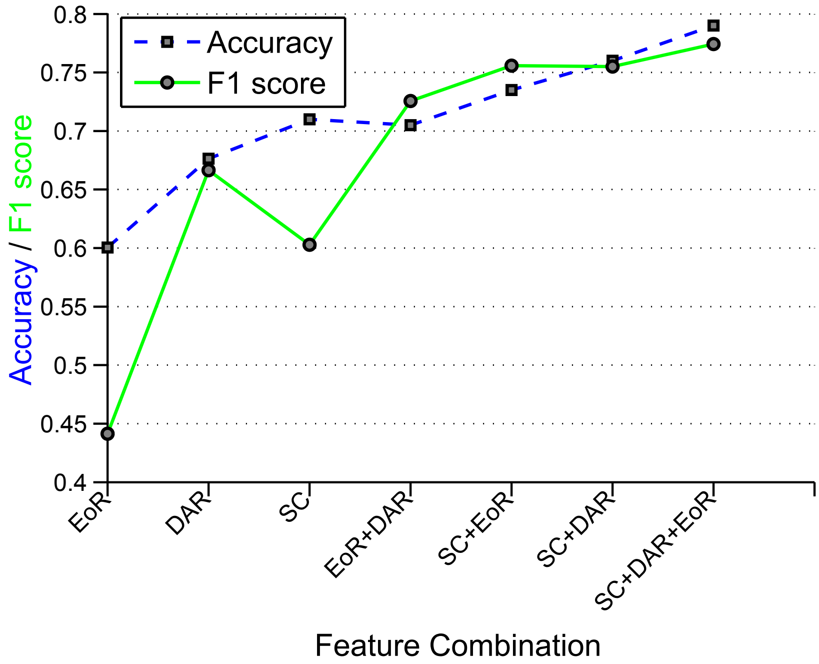


Figure 6.3: Accuracy and F1 score for different combinations of the features extracted from the DAR, EoR detection and situational complexity (SC).

low value whereas the accuracy exceeds 70%. This usually indicates a single-sided weighting of the classifier leading to a high recall of one class and to many false alarms for the other class. In this study, most of the take-overs in situation *Straight* are high-quality resulting in the single-sided weighting of this class and the high accuracy. When considering only driver-dependent features denoted with *EoR+DAR*, this single-sided weighting can be avoided. However, the continuously increasing F1 score and accuracy curve illustrate the obvious improvement of the classification for combinations of both traffic situation as well as driver-dependent features. Finally, the constant level of the F1 score and the simultaneous rise of the accuracy of only 3% between the combinations *SC+EoR* and *SC+DAR* indicate that the features of both driver-dependent information resources are of similar significance for the classifier.

6.3.2 Type of Intervention

To investigate the performance of the classifier with regard to the intervention type in the corresponding take-over situation, i.e. longitudinal or lateral intervention, Figure 6.4 shows the accuracy and F1 score of the classification based on the data for the situations *Straight* and *Braking* or *Straight* and *Cross-wind*. Again, the Ground Truth data is used for this evaluation. Clearly, the intervention type for the corresponding take-over situation has a significant impact on the classifier. Both the accuracy and F1 score are increased by ca. 12% and reach a value of about 87% in the situations with lateral intervention. The data

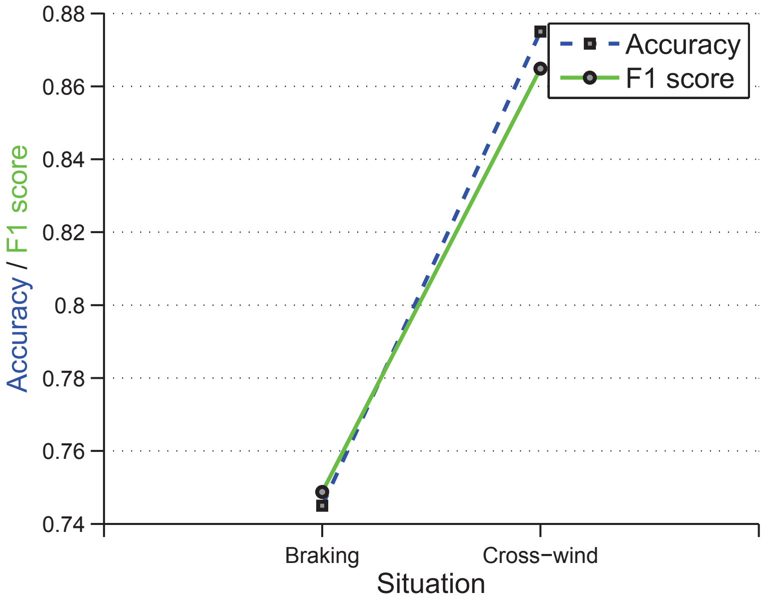


Figure 6.4: Classification accuracy and F1 score for the take-over situations *Braking* and *Cross-wind*.

indicates that distracted drivers were still able to manage the braking situation whereas the cross-wind situation on average posed a more challenging task. This phenomenon can be explained by considering the behavior of a distracted driver in take-over situations. Drivers tend to brake precautionarily in case of sudden emergency situations, especially when they are distracted. In the situation with the necessary lateral intervention, this behavior is of little or no actual use since the vehicle keeps drifting out of the lane. However, for the situation with the longitudinal intervention, braking is the correct behavior and therefore even extremely distracted drivers sometimes performed a reasonable take-over, resulting in the lower classification performance. This observation is in agreement with the findings by Zeeb et al. in [97].

6.3.3 Behavior of the ADAS

All results shown up to this point are based on the labelled Ground Truth data of the EoR gazes and DAR. However for the implementation of the proposed ADAS, automated methods for DAR and EoR detection are necessary. In the following, the algorithms described in Sections 4.4 and 5.3 are applied to generate the input for the classification of the take-over readiness with the linear SVM by detecting the EoR gazes and recognizing the current secondary task. As a consequence, the accuracy of the classifier is reduced by 9% to 70% and the F1 score decreases by 7% to 70%. While state-of-the-art methods for EoR detection usually provide a high accuracy of over 90%, current methods for DAR range from accuracies of 70% to 85%. However, the poor recorded gaze direction means the fallback strategy described in Section 4.4 had to be applied. Hence, the classification of the ADAS

Take-Over Quality	Algorithm issued a warning	Algorithm did not issue a warning
Low Take-Over Quality	38 (63%)	22 (37%)
High Take-Over Quality	24 (13%)	159 (87%)

Table 6.5: Behavior of the ADAS for the 60 s-intervals before each take-over situation.

was based on EoR features suffering from the reduced accuracy of the fallback strategy and DAR. As a result, the accuracy and F1 score of the classification decrease. Nevertheless, given the above mentioned conditions this is still a reasonable classification result especially with regard to the potential improvement by means of replacing the substandard gaze direction.

For each subject, a subject-independent classifier was learned and combined with the automated DAR, EoR detection, and classification of the situational complexity to the ADAS proposed in the previous sections. For the evaluation of this system, the driving simulator study was resimulated for the 60 s intervals before each take-over situation and used as input for the ADAS. Only in these situations can an evaluation of the correctness of a prompted warning be performed due to the occurring take-over. The driver's take-over readiness was classified for each sampling point of the 100 Hz signals in these intervals. If a low take-over quality was classified continuously for more than 2 s, the ADAS issued a warning. The threshold of 2 s was selected in reference to the design guidelines established by the NHTSA for in-vehicle electronic devices in [129]. Table 6.5 contains the absolute and relative number of drivers who received at least one warning in the 60 s interval before the take-over occurred; 63% of the drivers experiencing a low take-over situation would have received at least one warning during the 60 s interval prior to the take-over. On average, the warnings occurred 10 s before the take-over was prompted by the system. Since these drivers would have been warned shortly before the actual take-over, the ADAS holds the potential to prevent more than half of the critical situations. For the remaining 22 situations with a low take-over quality, the algorithm did not warn the driver preemptively. The ADAS was parameterized to minimize false alarms. With regard to the situations with a high take-over quality, 87% of the drivers would not have been interrupted by a warning whereas the false alarms occurred in the remaining 13% of the situations. This is an essential characteristic of an ADAS since the acceptance of such a system would significantly decrease if there were frequent false alarms. Moreover, the interval between two warnings averages 120 s and increases to 144 s if the ADAS is applied to the complete conditionally automated driving sections of each experiment which do not include further take-over situations. These intervals approximate the reported interval lengths of drowsiness detection systems in conditionally automated vehicles, although even longer intervals would be preferable [14].

6.4 Summary

In this chapter, the first known ADAS for an automated classification of the driver's take-over readiness in conditionally automated scenarios was introduced. To highlight the necessity for such systems, it was shown that there is scant literature available describing automated systems for the automated detection of the driver's take-over readiness. The proposed system incorporates the automated methods for Eyes-on-Road detection and driver-activity recognition of the previous Chapter 4 and Chapter 5 with an approach for determining the situational complexity. Based on the extracted features and a formal definition of the take-over quality, different classifier were trained and compared to each other with regard to their computational time and overall classification performance. The results highlight the preferred combination of a linear SVM with a feature selection step. Moreover, the different types of features were analyzed to prove that both driver-dependent and situation-dependent features are necessary to enable a high classification accuracy and are of similar significance for the classifier. Further, the analysis verified that the intervention type significantly impacts the classification performance. For take-over situations with longitudinal interventions, the accuracy of the classifier decreases since even inattentive drivers tend to brake precautionarily in case of sudden emergency situations, which by chance is the correct action to perform in this situation. Finally, the benefit of the ADAS is underscored by results showing that for 63% of the situations with a low take-over quality, the driver would have been warned shortly before the situation occurred. Moreover, nearly 90% of the drivers performing an adequate take-over would not have been interrupted by the system.

These evaluations were performed based on the thorough conditionally automated driving simulator study described in Subsection 2.4.4. Note that the evaluation of the system based on real-world driving scenarios is a challenging task since take-over situations may evolve to safety-critical situations. As a result, real-driving studies must contain only simple take-over situations or be conducted on test tracks which reduces the comparability to public roads. Another point to mention is that the proposed ADAS is a preemptive system, i.e., the system recognizes drivers with a low take-over readiness in relation to the complexity of the current traffic situation. However, no part of the ADAS is able to foresee an upcoming take-over situation. In fact, this is an extremely challenging task for automated systems and depends largely on the vehicle's sensors. Take-over situations due to reaching the end of a proper road, e.g., leaving the highway or entering a construction site, could be determined via accurate and up-to-date maps provided over vehicle communications. In case of sudden events such as an accident ahead of the vehicle, the take-over is triggered as soon as the sensors detect the event. Hence, an occurrence probability for take-over situations can hardly be incorporated into the ADAS. As a result, most of the prompted warnings will not be followed by an actual take-over situation. To some extent the assessment of the complexity of the traffic situation compensates for this weakness since the take-over situations in the conducted simulator study do not occur suddenly but rather develop gradually, e.g., the traffic density increases steadily before the situation Braking.

7 Summary and Future Work

In conditionally automated driving scenarios, take-over situations may occur which force the driver to reassume the control over and responsibility for the vehicle. The quality of such take-over situations is significantly influenced by various factors such as the traffic situation, the driver's gazes at the road, and the secondary task performed during the conditionally automated drive. Hence, the question arises how the take-over readiness of a driver could be individually recognized by an automated approach in the vehicle. Therefore, this work introduced the first approach enabling the classification of the driver's take-over readiness in conditionally automated driving scenarios. This approach is based on features extracted from the mentioned influence factors: traffic situation, secondary task, and the gazes at the road. Especially the recognition of the driver's activity and the detection of gazes at the road were at the focus of this work and can be realized by means of a driver monitoring approach with a driver camera. Both aspects require methods which can be applied online in a vehicle and reach sufficient accuracy even in the light of near-to-production camera systems. The results of this work show that a combination of features based on the traffic situation and driver monitoring performs best reaching an overall accuracy of 79%. However, the evaluation indicates that the type of intervention has a significant influence on the classification which is in accordance with the findings presented in the literature.

An ADAS based on such a classifier is not only able to determine the driver's take-over readiness, which could then be used by further applications to support the driver during the take-over. It can also be seen as an enabler for additional secondary tasks, since a driver with a low take-over readiness can be detected and could be kept in the loop through requested gazes at the road. Following this concept, a first prototype of such an ADAS was introduced and applied as a resimulation on the data of a driving simulator study. In case of the take-over situations with a low take-over quality, more than half of the drivers (63%) would have been prompted to perform gazes at the road due to the prototypical ADAS. On average, these prompts would have been triggered 10 s before the actual take-over, which might have enabled these drivers to perform it adequately. At the same time, 87% of the drivers performing a high-quality take-over would not have been interrupted by any warning indicating the low false detection rate.

For extracting features with regard to the driver's activity, methods for driver-activity recognition are necessary. However, known methods cannot distinguish between detailed secondary tasks such as reading or watching a video or are those applicable only in static lab environments. As a consequence, different methods for driver-activity recognition adapted to the conditionally automated driving environment were developed in this work. The most promising approach based on the driver's scanpath showed outstanding evaluation results for data of a driving simulator as well as for a real-driving study. Hence, the application of such methods in series vehicles will become possible in the near future.

Various approaches for extracting features with regard to the driver's gazes at the road have already been published in literature. Nevertheless, all of these approaches suffer from various shortcomings such as: required calibration steps, inability to detect moveable areas-of-interest, or absolute head pose as the fallback strategy in case of a deteriorated or missing gaze signal. To resolve these shortcomings, a novel approach based on dynamic clusters in space representing areas-of-interest was proposed. The clusters are represented by a mixture distribution and can be updated individually and in an online-fashion. Further, a fallback strategy able to detect gazes at the road by analyzing the driver's head movements was introduced. The cluster based approach showed nearly perfect detection performance in the real-driving study with a near-to-production driver camera.

Some of the methods introduced in this work for driver-activity recognition and detecting gazes at the road benefit from robust and accurate eye movement classification methods. Although this topic received much attention in the last decades, the science community is still divided on the use of threshold-based or probabilistic methods. Several studies claim that a fixed threshold is sufficient for the separation of fixations and saccades while others support methods able to adapt to varying conditions. However, no evaluation has yet been provided proving the actual existence of varying eye movement behavior, for example, in traffic situations. In this work, a thorough conditionally automated driving simulator study was evaluated with regard to the occurring variations during different secondary tasks. It was shown that the eye movement behavior does in fact vary significantly between these tasks and especially between idle and busy drivers. Moreover, current methods for eye movement classification adapt insufficiently to these variations. This motivated the creation of a novel approach called MERCY for eye movement classification with increased adaptability.

Hopefully, this work highlights the usefulness and maturity level of these approaches, especially of Eyes-on-Road systems with regard to an application in series vehicles, and will inspire further studies in the field of automated classification of the driver's take-over readiness and driver-activity recognition. The fact remains that although the technology is available, there are still far too few assistance systems on the market which focus on reducing the number of accidents due to a visual distracted drivers. By replacing the driver with an automated driving function in conditionally automated scenarios, such accidents could be prevented. However, it is still an open question if the driver might be overwhelmed in some of the take-over situations. Hence, some vehicle manufacturer already plan on skipping the introduction of conditionally automated driving functions on account of the discussed take-over scenarios. At this point, automated systems for classifying the driver's take-over readiness could close the gap and enable the first conditionally automated driving function within the next few years.

List of Figures

List of Tables

Bibliography

[1] Organisation Internationale des Constructeurs d'Automobiles (OICA). World motor vehicle production by country and type 2014-2015. http://www.oica.net/wp-content/uploads//Total-2015-Q4-March-16.pdf, 2016. Online Statistics; accessed 27-July-2016.

[2] Organisation Internationale des Constructeurs d'Automobiles (OICA). World vehicles in use 2005 - 2014 - all vehicles. http://www.oica.net/wp-content/uploads//total-inuse-2014.pdf, 2015. Online Statistics; accessed 27-July-2016.

[3] Deutscher Verkehrssicherheitsrat DVR. Unfallstatistik aktuell. http://www.dvr.de/betriebe_bg/daten/unfallstatistik/de_monate.htm. Online Statistics; accessed 27-July-2016.

[4] DEKRA Automobil GmbH. Verkehrssicherheitsreport 2015. Strategien zur Unfallvermeidung auf den Straßen Europas. Technical report, DEKRA Automobil GmbH, 2015.

[5] Daniel Damböck, Klaus Bengler, Mehdi Farid, and Lars Tönert. Übernahmezeiten beim hochautomatisierten Fahren. *Münchener Tagung Fahrerassistenz*, 2012.

[6] Natasha Merat, A Hamish Jamson, Frank CH Lai, Michael Daly, and Oliver MJ Carsten. Transition to manual: Driver behaviour when resuming control from a highly automated vehicle. *Transportation research part F: traffic psychology and behaviour*, 27:274–282, 2014.

[7] Ina Petermann-Stock, Linn Hackenberg, Tobias Muhr, and Christian Mergl. Wie lange braucht der Fahrer - Eine Analyse zu Übernahmezeiten aus verschiedenen Nebentätigkeiten während einer hochautomatisierten Staufahrt. *6. Tagung Fahrerassistenzsysteme. Der Weg zum automatischen Fahren*, 2013.

[8] Jonas Radlmayr, Christian Gold, Lutz Lorenz, Mehdi Farid, and Klaus Bengler. How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 58, pages 2063–2067. Sage Publications, 2014.

[9] Kathrin Zeeb, Axel Buchner, and Michael Schrauf. Is take-over time all that matters? The impact of visual-cognitive load on driver take-over quality after conditionally automated driving. *Accident Analysis & Prevention*, 92:230–239, 2016.

[10] Manuela Härtel. The impact of non-driving related tasks on the take-over performance from conditionally automated to manual driving: A driving suimulator study. Master's thesis, Otto-Friedrich-University Bamberg, 2015.

[11] Kathrin Zeeb, Axel Buchner, and Michael Schrauf. What determines the take-over time? An integrated model approach of driver take-over after automated driving. *Accident Analysis & Prevention*, 78:212–221, 2015.

[12] Christian Braunagel, David Geisler, Wolfgang Stolzmann, Wolfgang Rosenstiel, and Enkelejda Kasneci. On the necessity of adaptive eye movement classification in conditionally automated driving scenarios. In *ACM Symposium on Eye Tracking Research & Applications (ETRA 2016)*, 2016.

[13] Christian Braunagel, Till Prangel, Siemone Sieloff, and Wolfgang Stolzmann. Verfahren und Vorrichtung zur Erkennung einer Reaktionsfähigkeit eines Fahrers beim automatisierten Fahren. Patent DE 10 2015 001 686.5, issued 2015.

[14] Jürgen Schmidt, Christian Braunagel, Wolfgang Stolzmann, and Katja Karrer-Gauß. Driver drowsiness and behavior detection in prolonged conditionally automated drives. In *Intelligent Vehicles Symposium (IV 2016)*, pages 400–405. IEEE, 2016.

[15] Christian Braunagel, Enkelejda Kasneci, Wolfgang Stolzmann, and Wolfgang Rosenstiel. Driver-activity recognition in the context of conditionally autonomous driving. In *18th International IEEE Conference on Intelligent Transportation Systems (ITSC 2015)*, 2015.

[16] Christian Braunagel, Wolfgang Stolzmann, Enkelejda Kasneci, Thomas C Kübler, Wolfgang Fuhl, and Wolfgang Rosenstiel. Exploiting the potential of eye movements analysis in the driving context. In *15. Internationales Stuttgarter Symposium*, pages 1093–1105. Springer, 2015.

[17] Christian Braunagel, David Geisler, Wolfgang Rosenstiel, and Enkelejda Kasneci. Online recognition of driver-activity based on visual scanpath classification. *IEEE Intelligent Transportation Systems Magazine*, 9(4):23–36, 2017.

[18] Christian Braunagel, Wolfgang Rosenstiel, and Enkelejda Kasneci. Ready for Take-Over? A New Driver Assistance System for an Automated Classification of Driver Take-Over Readiness. *IEEE Intelligent Transportation Systems Magazine*, 9(4):10–22, 2017.

[19] Tom M Gasser, Clemens Arzt, Mihiar Ayoubi, Arne Bartels, Lutz Bürkle, Jana Eier, Frank Flemisch, Dirk Häcker, Tobias Hesse, Werner Huber, et al. Rechtsfolgen zunehmender Fahrzeugautomatisierung. *Berichte der Bundesanstalt für Straßenwesen. Unterreihe Fahrzeugtechnik*, (83), 2012.

[20] National Highway Traffic Safety Administration. Preliminary statement of policy concerning automated vehicles. Washington, DC, 2013.

[21] SAE International. J3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems, 2014.

[22] Daimler AG. Distronic plus with steering assist. https://techcenter.mercedes-benz.com/en/distronic_plus_steering_assist/detail.html. Product specification; accessed 12-August-2016.

[23] BMW AG. Steering and lane control assistant incl. traffic jam assistant. http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/driver_assistance/intelligent_driving.html#acc. Product specification; accessed 12-August-2016.

[24] Inc. Tesla Motors. Model s software version 7.0. https://www.tesla.com/presskit/autopilot. Product specification; accessed 12-August-2016.

[25] The Verge. Tesla driver killed in crash with Autopilot active, NHTSA investigating. http://www.theverge.com/2016/6/30/12072408/tesla-autopilot-car-crash-death-autonomous-model-s. Consumer Reports; accessed 12-August-2016.

[26] Lisanne Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983.

[27] Mica R Endsley. Automation and situation awareness. *Automation and human performance: Theory and applications*, pages 163–181, 1996.

[28] Richard S Snell and Michael A Lemp. Clinical anatomy of the eye. John Wiley & Sons, 2013.

[29] AL Yarbus. Eye movements and vision. *New York*, 1967.

[30] Gustav Osterberg. Topography of the layer of rods and cones in the human retina. Nyt Nordisk Forlag, 1935.

[31] Michael F Land. The human eye: Structure and function. *Nature Medicine*, 5(11):1229–1229, 1999.

[32] David A Goss and Roger W West. Introduction to the Optics of the Eye. Butterworth-Heinemann Medical, 2001.

[33] Genes-Vision Foundation CH. Eye anatomy. https://www.genes-vision.ch/retinalearn/eye-anatomy/, 2016. Online Description; accessed 18-August-2016.

[34] R John Leigh and David S Zee. The neurology of eye movements. Oxford University Press, 2015.

[35] Enkelejda Kasneci. Towards the automated recognition of assistance need for drivers with impaired visual field. PhD thesis, Universität Tübingen, 2013.

[36] Manfred Schweigert. Fahrerblickverhalten und Nebenaufgaben. PhD thesis, Universität München, 2003.

[37] Tony Matthias Poitschke. Blickbasierte Mensch-Maschine Interaktion im Automobil. PhD thesis, Universität München, 2011.

[38] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[39] Matthias Höffken, Emin Tarayan, Ulrich Kreßel, and Klaus Dietmayer. Stereo vision-based driver head pose estimation. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 253–260. IEEE, 2014.

[40] Manuel Schäfer, Emin Tarayan, and Ulrich Kreßel. Robust facial landmark localization for automotive applications. In *Advanced Microsystems for Automotive Applications 2016*, pages 91–102. Springer, 2016.

[41] David Cristinacce and Timothy F Cootes. Facial feature detection and tracking with automatic template selection. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 429–434. IEEE, 2006.

[42] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models - their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[43] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[44] Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, et al. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[45] Ascension Technology Corporation. laserBird - Installation and Operation Guide, 2004.

[46] Laurence R Young and David Sheena. Survey of eye movement recording methods. *Behavior research methods & instrumentation*, 7(5):397–429, 1975.

[47] David A Robinson. A method of measuring eye movemnent using a scieral search coil in a magnetic field. *IEEE Transactions on bio-medical electronics*, 10(4):137–145, 1963.

[48] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. Eye tracking: A comprehensive guide to methods and measures. Oxford University Press, 2011.

[49] Elias Daniel Guestrin and Moshe Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133, 2006.

[50] Ergoneers GmbH, Manching. Dikablis - Das Blickerfassungssystem. Benutzerhandbuch Dikablis Software, 2011.

[51] Ergoneers GmbH, Manching. Dikablis Professional - Handbuch, 2015.

[52] Thomas C Kübler, Colleen Rothe, Ulrich Schiefer, Wolfgang Rosenstiel, and Enkelejda Kasneci. Subsmatch 2.0: Scanpath comparison and classification based on subsequence frequencies. *Behavior Research Methods*, pages 1–17, 2016.

[53] Eberhard Zeeb. Daimler's new full-scale, high-dynamic driving simulator–a technical overview. In *Conference Proc. Driving Simulator Conference Europe, Paris*, 2010.

[54] Bob Kuehne and Sean Carmody. Design of a modern image generation engine for driving simulation. *Actes INRETS*, pages 259–266, 2010.

[55] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78. ACM, 2000.

[56] Enkelejda Kasneci, Gjergji Kasneci, Thomas C Kübler, and Wolfgang Rosenstiel. The applicability of probabilistic methods to the online recognition of fixations and saccades in dynamic scenes. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 323–326. ACM, 2014.

[57] Heino Widdel. Operational problems in analysing eye movements. *Advances in Psychology*, 22:21–29, 1984.

[58] Joseph H Goldberg and Jack C Schryver. Eye-gaze-contingent control of the computer interface: Methodology and example for zoom detection. *Behavior research methods, instruments, & computers*, 27(3):338–350, 1995.

[59] Dario D Salvucci and John R Anderson. Tracing eye movement protocols with cognitive process models. 1998.

[60] D Sauter, BJ Martin, N Di Renzo, and C Vomscheid. Analysis of eye tracking movements using innovations generated by a kalman filter. *Medical and biological Engineering and Computing*, 29(1):63–69, 1991.

[61] Oleg V Komogortsev and Javed I Khan. Eye movement prediction by kalman filter with integrated linear horizontal oculomotor plant mechanical model. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 229–236. ACM, 2008.

[62] Enkelejda Tafaj, Gjergji Kasneci, Wolfgang Rosenstiel, and Martin Bogdan. Bayesian online clustering of eye movement data. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 285–288. ACM, 2012.

[63] Enkelejda Tafaj, Thomas C Kübler, Gjergji Kasneci, Wolfgang Rosenstiel, and Martin Bogdan. Online classification of eye tracking data for automated analysis of traffic hazard perception. In *Artificial Neural Networks and Machine Learning–ICANN 2013*, pages 442–450. Springer, 2013.

[64] Casper J Erkelens and Ingrid MLC Vogels. The initial direction and landing position of saccades. *Studies in Visual Information Processing*, 6:133–144, 1995.

[65] Tayyar Sen and Ted Megaw. The effects of task variables and prolonged performance on saccadic eye movement parameters. *Advances in Psychology*, 22:103–111, 1984.

[66] Matthias Rötting. Parametersystematik der Augen-und Blickbewegungen für arbeitswissenschaftliche Untersuchungen. Shaker, 2001.

[67] Monica S Castelhano and John M Henderson. Stable individual differences across images in human saccadic eye movements. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 62(1):1, 2008.

[68] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124(3):372, 1998.

[69] John M Winn and Christopher M Bishop. Variational message passing. In *Journal of Machine Learning Research*, pages 661–694, 2005.

[70] MF Benjamin. Miller-keane encyclopedia and dictionary of medicine, nursing and allied health. Philadelphia: Saunders, 1997.

[71] Markus Dahm. Grundlagen der Mensch-Computer-Interaktion. Pearson Studium München, 2006.

[72] Michael Sivak. The information that drivers use: is it indeed 90% visual? *Perception*, 25(9):1081–1089, 1996.

[73] Thomas A Dingus, Sheila Garness Klauer, VL Neale, A Petersen, SE Lee, JD Sudweeks, MA Perez, J Hankey, DJ Ramsey, S Gupta, et al. The 100-car naturalistic driving study, phase ii-results of the 100-car field experiment. Technical report, NHTSA, 2006.

[74] Walter W Wierwille and Louis Tijerina. Modelling the relationship between driver in-vehicle visual demands and accident occurrence. *Vision in vehicles*, 6:233–243, 1998.

[75] A Hamish Jamson and Natasha Merat. Surrogate in-vehicle information systems and driver behaviour: Effects of visual and cognitive load in simulated rural driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8(2):79–96, 2005.

[76] Jeff Greenberg, Louis Tijerina, Reates Curry, Bruce Artz, Larry Cathey, Dev Kochhar, Ksenia Kozak, Mike Blommer, and Peter Grant. Driver distraction: Evaluation with event detection paradigm. *Transportation Research Record: Journal of the Transportation Research Board*, (1843):1–9, 2003.

[77] Brendan Wallace. External-to-vehicle driver distraction. Scottish Executive, Social Research, 2003.

[78] Kang-chen Chen and Hye Jung Choi. Visual attention and eye movements. http://www.ics.uci.edu/~majumder/vispercep/paper08/visualattention.pdf, 2008. Online Lecture Note; accessed 21-June-2016.

[79] Alexandra Frischen, Andrew P Bayliss, and Steven P Tipper. Gaze cueing of attention: visual attention, social cognition, and individual differences. *Psychological bulletin*, 133(4):694, 2007.

[80] Peter R Chapman and Geoffrey Underwood. Visual search of driving situations: Danger and experience. *Perception*, 27(8):951–964, 1998.

[81] Heikki Summala, Tapio Nieminen, and Maaret Punto. Maintaining lane position with peripheral vision during in-vehicle tasks. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 38(3):442–451, 1996.

[82] Panos Konstantopoulos, Peter Chapman, and David Crundall. Driver's visual attention as a function of driving experience and visibility. Using a driving simulator to explore drivers' eye movements in day, night and rain driving. *Accident Analysis & Prevention*, 42(3):827–834, 2010.

[83] Yiyun Peng, Linda Ng Boyle, Mahtab Ghazizadeh, and John D Lee. Factors affecting glance behavior when interacting with in-vehicle devices: implications from a simulator study. In *7th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*. Bolton Landing, NY, 2013.

[84] William Horrey and Christopher Wickens. In-vehicle glance duration: distributions, tails, and model of crash risk. *Transportation Research Record: Journal of the Transportation Research Board*, (2018):22–28, 2007.

[85] Trent W Victor, Joanne L Harbluk, and Johan A Engström. Sensitivity of eye-movement measures to in-vehicle task difficulty. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8(2):167–190, 2005.

[86] Ashish Tawari, Kuo Hao Chen, and Mohan M Trivedi. Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 988–994. IEEE, 2014.

[87] Kenichi Ohue, Yukinori Yamada, Shigeyasu Uozumi, Setsuo Tokoro, Akira Hattori, and Takeshi Hayashi. Development of a new pre-crash safety system. Technical report, SAE Technical Paper, 2006.

[88] Katja Kircher, Albert Kircher, and Fredrich Claezon. Distraction and drowsiness–a field study. *Linköping, Sweden: VTI, Swedish National Road and Transport Research Institute*, 2009.

[89] Francisco Vicente, Zehua Huang, Xuehan Xiong, Fernando De la Torre, Wende Zhang, and Dan Levi. Driver gaze tracking and eyes off the road detection system. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2014–2027, 2015.

[90] Borhan Vasli, Sujitha Martin, and Mohan Manubhai Trivedi. On driver gaze estimation: Explorations and fusion of geometric and data driven approaches. In *19th International Conference on Intelligent Transportation Systems (ITSC 2016)*, pages 655–660. IEEE, 2016.

[91] Paul Smith, Mubarak Shah, and Niels da Vitoria Lobo. Determining driver visual attention with one camera. *IEEE transactions on intelligent transportation systems*, 4(4):205–218, 2003.

[92] Benjamin Trefflich. Videogestützte Überwachung der Fahreraufmerksamkeit und Adaption von Fahrerassistenzsystemen. PhD thesis, Technische Universität Ilmenau, 2010.

[93] Lars Galley, Elisabeth Hendrika Hentschel, Klaus-Peter Kuhn, and Wolfgang Stolzmann. Verfahren und Steuergerät zum fahrerindividuellen Erkennen von Unaufmerksamkeiten des Fahrers eines Fahrzeuges. Patent DE 10 2005 026 457 A1, issued 2006.

[94] Parisa Ebrahim. Driver drowsiness monitoring using eye movement features derived from electrooculography. PhD thesis, Universität Stuttgart, 2016.

[95] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[96] Esteban Gutierrez Mlot, Hamed Bahmani, Siegfried Wahl, and Enkelejda Kasneci. 3d gaze estimation using eye vergence. *9th International Conference on Health Informatics, Healthinf 2016*, 02 2016.

[97] K Zeeb, M Härtel, A Buchner, and M Schrauf. Why is steering not the same as braking? The impact of secondary task load on lateral and longitudinal driver interventions during conditionally automated driving. *Transportation Res. F: Traffic Psychol. Behaviour*, 2016.

[98] Hua Zhong, Jianbo Shi, and Mirkó Visontai. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[99] Ilkka Korhonen, Juha Parkka, and Mark Van Gils. Health monitoring in the home of the future. *IEEE Engineering in medicine and biology magazine*, 22(3):66–73, 2003.

[100] Alex Pentland. Smart rooms, smart clothes. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 949–953. IEEE, 1998.

[101] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[102] Eshed Ohn-Bar and Mohan M Trivedi. Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns. In *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1245–1250. IEEE, 2014.

[103] Sangho Park and Mohan Trivedi. Driver activity analysis for intelligent vehicles: issues and development framework. In *Intelligent Vehicles Symposium*, pages 644–649. IEEE, 2005.

[104] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *Communications Surveys & Tutorials, IEEE*, 15(3):1192–1209, 2013.

[105] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[106] Uwe Maurer, Asim Smailagic, Daniel P Siewiorek, and Michael Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks*, pages 4–pp. IEEE, 2006.

[107] Amardeep Sathyanarayana, Sandhya Nageswaren, Hassan Ghasemzadeh, Roozbeh Jafari, and John HL Hansen. Body sensor networks for driver distraction identification. In *IEEE International Conference on Vehicular Electronics and Safety (ICVES 2008)*, pages 120–125. IEEE, 2008.

[108] Walter W Wierwille, SS Wreggit, CL Kirn, LA Ellsworth, and RJ Fairbanks. Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness. final report. Technical report, 1994.

[109] Johan Engström, Emma Johansson, and Joakim Östlund. Effects of visual and cognitive load in real and simulated motorway driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 8(2):97–120, 2005.

[110] Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Troster. Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):741–753, 2011.

[111] Anwesha Banerjee, Shreyasi Datta, Amit Konar, DN Tibarewala, and Janarthanan Ramadoss. Cognitive activity recognition based on electrooculogram analysis. In *Advanced Computing, Networking and Informatics-Volume 1*, pages 637–644. Springer, 2014.

[112] Yuki Shiga, Andreas Dengel, Takumi Toyama, Koichi Kise, and Yuzuko Utsumi. Daily activity recognition combining gaze motion and visual features. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1103–1111. ACM, 2014.

[113] Shoya Ishimaru, Kai Kunze, Koichi Kise, Jens Weppner, Andreas Dengel, Paul Lukowicz, and Andreas Bulling. In the blink of an eye: combining head motion and eye blink frequency for activity recognition with google glass. In *Proceedings of the 5th Augmented Human International Conference*, page 15. ACM, 2014.

[114] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. BioNumbers - the database of key numbers in molecular and cell biology. *Nucleic acids research*, 38(suppl 1):D750–D753, 2010.

[115] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[116] Christopher M Bishop et al. Pattern recognition and machine learning, volume 4. Springer New York, 2006.

[117] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[118] Thomas C Kübler, Enkelejda Kasneci, and Wolfgang Rosenstiel. Subsmatch: Scanpath similarity in dynamic scenes based on subsequence frequencies. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 319–322. ACM, 2014.

[119] ISO. Iso14198: Road vehicles - ergonomic aspects of transport information and control systems - calibration tasks for methods which asses driver demand due to the use of in-vehicle systems, 2012.

[120] Anna Feldhütter, Christian Gold, Sonja Schneider, and Klaus Bengler. How the duration of automated driving influences take-over performance and gaze behavior. pages 309–318, 2017.

[121] Frank Flemisch, Matthias Heesen, Tobias Hesse, Johann Kelsch, Anna Schieben, and Johannes Beller. Towards a dynamic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations. *Cognition, Technology & Work*, 14(1):3–18, 2012.

[122] Christian Gold, Lutz Lorenz, Daniel Damböck, and Klaus Bengler. Partially automated driving as a fallback level of high automation. 6. *Tagung Fahrerassistenzsysteme. Der Weg zum automatischen Fahren*, 28(29.11):2013, 2013.

[123] Josef Nilsson, Paolo Falcone, and Jonny Vinter. Safe transitions from automated to manual driving using driver controllability estimation. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1806–1816, 2015.

[124] Corp. Omron. OMRON Develops World's First Onboard Sensor Featuring Cutting-Edge AI. `http://www.omron.com/media/press/2016/06/c0606.html`. Global News; accessed 07-February-2017.

[125] H v Benda. Die Skalierung der Gefährlichkeit von Verkehrssituationen. I. Teil: Ein Klassifikationsschema für Verkehrssituationen aus Fahrersicht. FP 7320 im Auftrag der Bundesanstalt für Straßenwesen. München: Technische Universität, Lehrstuhl für Psychologie, 1977.

[126] Wolfgang Fastenmeier et al. Autofahrer und Verkehrssituation. Neue Wege zur Bewertung von Sicherheit und Zuverlässigkeit moderner Straßenverkehrssysteme. Number 33. 1995.

[127] Ekkehart Frieling and C Graf Hoyos. Fragebogen zur Arbeitsanalyse (FAA). H. Huber Bern, 1978.

[128] George Forman and Ira Cohen. Learning from little: Comparison of classifiers given little training. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 161–172. Springer, 2004.

[129] NHTSA. Visual-Manual NHTSA Driver Distraction Guidelines For In-Vehicle Electronic Devices. Technical report, 2010.