3D Human Body Pose-Based Activity Recognition for Driver Monitoring Systems

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Dipl.-Inform.

Manuel Martin

aus Heidelberg

Tag der mündlichen Prüfung:	02.02.2023
Erster Gutachter:	Prof. Dr.-Ing. Rainer Stiefelhagen
Zweiter Gutachter:	Prof. Dr.-Ing. J. Marius Zöllner

Abstract

Traffic accidents are unfortunately an unavoidable part of our modern transport system. In many cases the behavior of drivers is a contributing factor. Different studies show that distractions caused by secondary activities, like the use of mobile phones, play a large role. In addition, the trend to occupy oneself with other activities rises with increasing vehicle automation because drivers are less involved in the driving task. For this reason, driver monitoring systems will be required for future automated cars.

In this thesis we therefore investigate how to detect the activities of drivers in automated cars using a modular recognition system based on 3D data. The proposed system consists of two stages. The first stage creates a 3D interior state model from camera data including the 3D body pose of the driver, the location of elements of the interior as well as the position of objects involved in certain activities. The second stage uses this representation to detect activities. We base this design on two hypotheses: First, a rich 3D interior state model including other elements in addition to the 3D driver body pose is important to discern fine-grained activities. Second, the interior state model, created by the first stage, is a sensor modality and sensor location independent representation allowing the second stage to successfully detect activities even with large changes to the camera system. To verify these assumptions, we contribute to the research field in three areas.

The foundation of all our efforts are annotated datasets. Based on our extensive literature review we show that there are no suitable public automotive data sources. We therefore collect multiple datasets for different subtasks leading to our final dataset, published under the name Drive&Act. It includes a large-scale hierarchical activity recognition benchmark with multiple 3D input modalities for the task, like the 3D body pose of the driver, the location of interior elements like the steering wheel, as well as 3D trajectories of objects like smartphones. In addition, the dataset includes a public benchmark for 3D driver body pose estimation with challenging partial occlusion of the driver's body.

The main feature of the 3D interior state model is the body pose of the driver. Here we contribute methods for real-time 3D driver body pose estimation based on depth images. The primary challenge of using depth data for this task is occlusion of body parts. A second challenge is the lack of public automotive datasets to train these methods. In our first approach we therefore rely on simulated depth images with automated annotation for training. Our second approach handles this problem on the algorithm level. It uses a novel split of 2D body pose estimation followed by separate 3D keypoint regression guided by the depth image. This enables us to rely on advances in 2D body pose estimation using large-scale datasets from other domains. We can demonstrate the robustness of this method to partial occlusion on the 3D body pose benchmark of Drive&Act.

While the primary input of many related activity recognition methods is just the 3D human body pose, we research how to expand the input to a complex 3D state model including elements of the surrounding vehicle interior as well as positions of objects relevant for certain activities. We follow one central paradigm and assume that the distance in 3D space of keypoints of the driver's body to other elements in the state model is an important indicator of their relevance for the performed activity. Based on this hypothesis we develop different methods with increasing complexity of the interior state model. Our final method casts all parts of the interior state model into a spatio-temporal graph. To generate this graph, we rely on the distance of keypoints of the driver's body to other parts of the state model to determine which nodes to include in the graph and what edges to create. We analyze this graph using a neural network based on graph convolutions. We can show the advantage of our graph creation method in selecting relevant interior elements and objects and the usefulness of object location data to discern activities represented by similar body poses of the driver. Consequently, we can prove our initial hypothesis that additional input modalities improve the detection of fine-grained activities based on 3D data and we can quantify their impact.

We also investigate the overall performance of our modular system regarding sensor modality and viewpoint changes. We can demonstrate the capability to switch between creating the interior state model based on a multi-view camera system and creating it using data from a single depth sensor. We can show that our activity recognition approach can be trained on one of these representations and evaluated on the other with just a moderate performance drop. In addition, the overall system can generalize across different datasets recorded in different vehicles and in vastly different conditions, switching between data recorded in a simulator for automated driving and data recorded on a test track driving manually. This supports our second hypothesis that the 3D interior state model resulting from our first stage of algorithms is sensor independent to a large degree.

Kurzdarstellung

Verkehrsunfälle sind leider ein unvermeidbarer Bestandteil unseres modernen Verkehrssystems. In vielen Fällen trägt das Verhalten der Fahrer zum Unfall bei. Verschiedene Studien zeigen, dass Ablenkung durch Nebentätigkeiten, wie die Nutzung von Smartphones, eine große Rolle spielt. Die Tendenz sich mit anderen Tätigkeiten zu beschäftigen steigt mit zunehmender Automatisierung, da die Fahrer weniger in die Fahraufgabe eingebunden sind. Aus diesem Grund werden für zukünftige automatisierte Fahrzeuge Fahrerüberwachungssysteme benötigt.

In dieser Arbeit wird daher untersucht, wie die Aktivität von Fahrern in automatisierten Fahrzeugen mit Hilfe eines modularen Erkennungssystems auf der Grundlage von 3D-Daten erkannt werden kann. Die Erfassung läuft in zwei Schritten ab. Im ersten Schritt wird aus Kameradaten ein 3D-Zustandsmodell des Innenraums erstellt das die Körperpose des Fahrers, die Position von Elementen des Innenraums sowie die Position von Objekten, die an bestimmten Aktivitäten beteiligt sind, enthält. Im zweiten Schritt wird diese Darstellung zur Erkennung von Aktivitäten genutzt. Dieses Vorgehen stützt sich auf zwei Hypothesen: Erstens ist ein umfangreiches 3D-Zustandsmodell, das neben der 3D-Körperpose des Fahrers auch andere Elemente enthält, wichtig, um feingranular Aktivitäten zu erkennen. Zweitens ist das im ersten Schritt erstellte Zustandsmodell eine Repräsentation die sensorunabhängig ist und es der zweiten Stufe ermöglicht, selbst bei großen Veränderungen des Kamerasystems erfolgreich Aktivitäten zu erkennen. Um diese Annahmen zu verifizieren, leisten wir in drei Bereichen Beiträge zum Forschungsfeld.

Die Grundlage all unserer Verfahren sind annotierte Datensätze. Basierend auf unserer umfangreichen Literaturrecherche zeigen wir, dass es keine geeigneten öffentlichen Datenquellen für dieses Konzept gibt. Wir sammeln daher mehrere Datensätze für verschiedene Teilprobleme, die zu unserem endgültigen Datensatz führen, der unter dem Namen Drive&Act veröffentlicht wurde. Er umfasst einen hierarchischen Aktivitätserkennungs-Benchmark mit mehreren 3D-Eingabemodalitäten für die Aufgabe, wie die Körperpose des Fahrers, die Position von Innenraumelementen wie dem Lenkrad, sowie Trajektorien von Objekten wie Smartphones. Darüber hinaus enthält der Datensatz einen Benchmark für die 3D-Körperposenschätzung des Fahrers auch bei Teilverdeckung des Körpers.

Das Hauptmerkmal des 3D-Zustandsmodells ist die Körperpose des Fahrers. Wir tragen Methoden zur Echtzeitschätzung der 3D-Körperpose auf der Grundlage von Tiefenbildern bei. Die größte Herausforderung bei der Verwendung von Tiefenbildern für diese

Aufgabe ist der Umgang mit Verdeckungen von Körperteilen. Eine zweite Herausforderung ist der Mangel an öffentlichen Fahrzeugdatensätzen zum Trainieren dieser Methoden. Unser erster Ansatz stützt sich daher auf synthetisch generierte Trainingsdaten. Unser zweiter Ansatz behandelt dieses Problem algorithmisch. Er verwendet eine neuartige Aufteilung in 2D-Körperposenschätzung, gefolgt von einer separaten Regression der 3D-Position auf Basis des Tiefenbilds. Dadurch können wir uns auf Fortschritte bei der 2D-Körperposenschätzung stützen und große Datensätze aus anderen Bereichen für das Training verwenden. Wir können die Robustheit dieser Methode gegenüber teilweiser Verdeckung von Körperteilen anhand des Drive&Act Datensatzes zeigen.

Während die primäre Eingabe vieler verwandter Aktivitätserkennungsmethoden nur die 3D-Körperpose umfasst, erforschen wir, wie die Eingabe auf ein komplexes 3D-Zustandsmodell des Innenraums erweitert werden kann, das sowohl Elemente des umgebenden Fahrzeuginnenraums als auch die Position von Objekten enthält. Wir folgen einem zentralen Paradigma und gehen davon aus, dass der Abstand von Körperteilen des Fahrers zu anderen Elementen im Zustandsmodell ein wichtiger Indikator für deren Relevanz ist. Unser letzter Ansatz erstellt aus dem Zustandsmodell einen Graphen der sowohl die räumliche Position als auch die zeitliche Entwicklung der Einzelkomponenten abbildet. Für die Erstellung des Graphen verwenden wir den Abstand von Körperteilen des Fahrers zu anderen Teilen des Zustandsmodells, um zu bestimmen, welche Komponenten in den Graphen aufgenommen und welche Kanten erstellt werden. Der Graph wird anschließend mit auf graph convolution basierenden neuronalen Netzen analysiert. Wir können den Vorteil unseres Ansatzes zur Erstellung des Graphen bei der Auswahl relevanter Elemente anhand des Drive&Act Datensatzes zeigen. Weiterhin können wir durch die Analyse verschiedener Eingabemodalitäten auf dem Datensatz deren Nützlichkeit bestimmen und folglich unsere erste initiale Hypothese zur Nützlichkeit weiterer Eingabemodalitäten bestätigen.

Wir untersuchen auch die Gesamtleistung unseres modularen Systems in Bezug auf Wechsel des Sensortyps und des Kamerablickwinkels. Wir erstellen hierfür das 3D-Zustandmodell sowohl auf Basis eines Multi-Kamera Systems als auch auf Basis eines einzelnen Tiefensensors. Wir können zeigen, dass unser Aktivitätserkennungsansatz auf einer dieser Repräsentationen trainiert und auf der anderen mit nur geringem Leistungsabfall evaluiert werden kann. Darüber hinaus demonstrieren wir, dass das Gesamtsystem auch über verschiedene Datensätze hinweg verwendet werden kann, die in verschiedenen Fahrzeugen und unter sehr unterschiedlichen Bedingungen aufgezeichnet wurden. Mit diesen Experimenten können wir die Robustheit unseres Ansatzes bezüglich Veränderungen des Sensorsystem, und somit auch unsere zweite initiale Hypothese, nachweisen.

Acknowledgements

First, I would like to thank Prof. Dr.-Ing. Rainer Stiefelhagen for his mentoring and guidance. Especially the summer seminars helped facilitate scientific exchange, frequently led to new research ideas and were enjoyable in general. I would also like to thank the members of the CV:HCI lab for numerous insightful discussions that occurred during my visits. The reading group also helped me familiarize myself with new concepts and allowed me to gain knowledge across different areas of research. I am grateful to my colleagues at Fraunhofer IOSB for their support and their patience during the final years of my thesis. This project would not have been possible without the great collaboration with project partners in the research projects PAKoS and InCarIn. Finally, I would also like to thank my family and friends for their support and the occasional distraction from working on this thesis.

Contents

Abstract . . . . **i**

Kurzdarstellung . . . . **iii**

Acknowledgements . . . . **v**

1 Introduction **1**
1.1 The Driver in Automated Vehicles 2
1.2 Driver Monitoring for Takeover Scenarios 4
1.3 Applications for Driver Monitoring 7
1.4 Contributions and Outline 9

2 Related Work **13**
2.1 General Human Body Pose Estimation 13
2.1.1 Datasets 13
2.1.2 2D Human Body Pose Estimation 16
2.1.3 3D Human Body Pose Estimation from Depth Images 18
2.1.4 Other 3D Human Body Pose Estimation Methods 21
2.2 Driver Body Pose Estimation 22
2.2.1 Datasets 23
2.2.2 Methods 26
2.3 General Activity Recognition 28
2.3.1 Datasets 28
2.3.2 Video-Based Methods 29
2.3.3 3D Human Body Pose-Based Methods 30
2.4 Driver Activity Recognition 34
2.4.1 Datasets 35
2.4.2 Methods to Detect Interaction With the Interior 38
2.4.3 Image-Based Methods 39
2.4.4 Video-Based Methods 40
2.4.5 Driver Body Pose-Based Methods 41
2.5 Summary 43

3 Datasets for Driver Monitoring 45
3.1 Camera Selection for 3D Cabin Monitoring 45
3.2 Evaluation of Camera Views for Cabin Monitoring 50
3.3 Driver Depth Pose Dataset 52
3.3.1 Data Collection 53
3.3.2 Annotation 54
3.4 InCarIn Activity Dataset 55
3.4.1 Recording Environment 55
3.4.2 Pilot Study - InCarIn Interior Interaction Dataset 56
3.4.3 Data Collection 57
3.4.4 Activity Annotation 58
3.4.5 Interior Annotation 58
3.4.6 Summary 60
3.5 Drive&Act Dataset 60
3.5.1 Recording Environment 61
3.5.2 Data Collection 64
3.5.3 Activity Annotation 67
3.5.4 3D Driver Body Pose Annotation 69
3.5.5 Interior Annotation 70
3.5.6 Bounding Box Annotation 70
3.5.7 Evaluation Metrics 73
3.5.8 Summary 75

4 3D Driver Body Pose Estimation 77
4.1 Challenges for 3D Body Pose Estimation Using Depth Cameras 77
4.2 Random Decision Forests for 3D Driver Body Pose Estimation 79
4.2.1 Method 79
4.2.2 Random Decision Forest Training 82
4.2.3 Implementation Details 86
4.2.4 Evaluation on the Driver Depth Pose Dataset 87
4.3 DepthFix: Driver Body Pose Estimation with Occlusion Handling . . . 92
4.3.1 DepthFix method 93
4.3.2 Baseline: Multi-View Triangulation 95
4.3.3 Implementation Details 97
4.4 Evaluation on the Drive&Act Driver Body Pose Benchmark 98
4.4.1 OpenPose Performance for 2D Driver Body Pose Estimation . . 98
4.4.2 Random Decision Forests Trained on Synthetic Data 99
4.4.3 DepthFix Performance 100
4.5 Summary 106

5 Driver Activity Recognition **107**
5.1 Interior Interaction Detection 108
5.1.1 Method 108
5.1.2 Evaluation on the InCarIn Interior Interaction Dataset 110
5.2 Three-Stream Recurrent Driver Activity Recognition 112
5.2.1 Recurrent Networks 112
5.2.2 Three-Stream Recurrent Neural Network 114
5.2.3 Implementation Details 116
5.2.4 Evaluation on the InCarIn Activity Dataset 118
5.3 Dynamic Interaction Graphs 119
5.3.1 Spatio-Temporal Graph Convolution 120
5.3.2 Dynamic Interaction Graphs for Activity Recognition 122
5.3.3 Interaction Graph Network 125
5.3.4 Implementation Details 126
5.4 Evaluation on the Drive&Act Dataset 127
5.4.1 Impact of the Threshold for Creating Interaction Graphs 127
5.4.2 Performance of Our Own Methods on Drive&Act 129
5.4.3 Comparison to Related Methods 134
5.4.4 Cross-View and Cross-Modal Evaluation 135
5.5 Summary 140

6 Conclusion **141**
6.1 Future Work and Open Research Questions 143
6.2 Applications to Other Fields 144

Bibliography **145**

Own publications **179**

Appendix

A Simulated camera views **181**

B Interaction Graphs: Confusion Matrices for Drive&Act Using Different Input Modalities. **182**

1 Introduction

The focus of this thesis is driver activity recognition for automated cars. We research how this task can be solved in a modular way with a focus on 3D data and flexibility with regards to sensor systems and sensor placement. We believe that this approach results in a rich representation of the interior that allows to expand the system efficiently for future applications. To achieve this, our proposed approach consists of two stages: Generating a sensor-independent 3D feature representation of the interior, including the driver, followed by activity recognition.

The main feature of the camera-independent representation is the 3D body pose of the driver. Here we contribute 3D driver body pose estimation methods, based on data from depth cameras. In addition, we enrich the representation with an interior model of the vehicle as well as 3D positions of objects used for different activities.

Our proposed activity recognition system uses this representation as input. We investigate how to integrate the interior model as well as 3D object positions in addition to the 3D body pose of the driver.

In order to develop these approaches suitable datasets are necessary which were also collected and published as part of this thesis.

Figure 1.1 depicts an overview of our work. We motivate this approach with an overview of driver behavior in automated vehicles and driver monitoring applications before depicting our contributions and the thesis outline in detail.

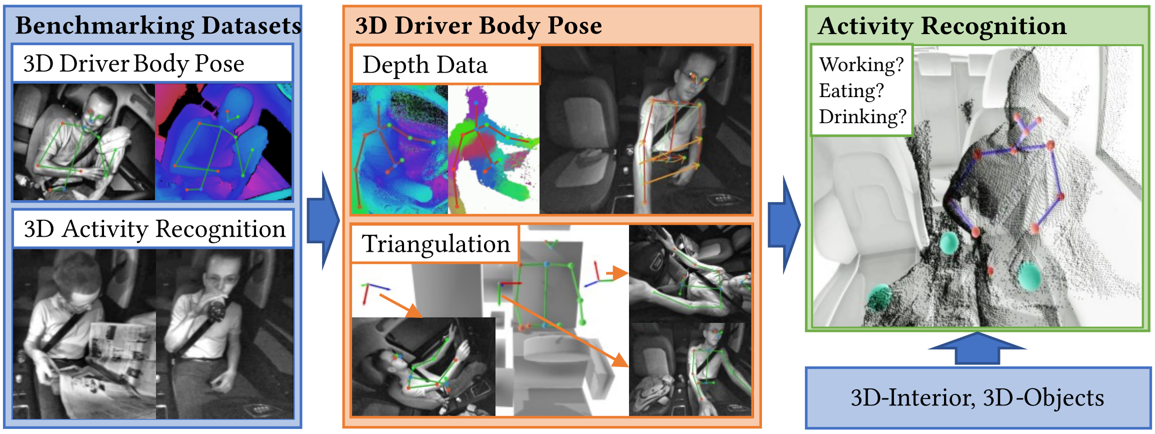


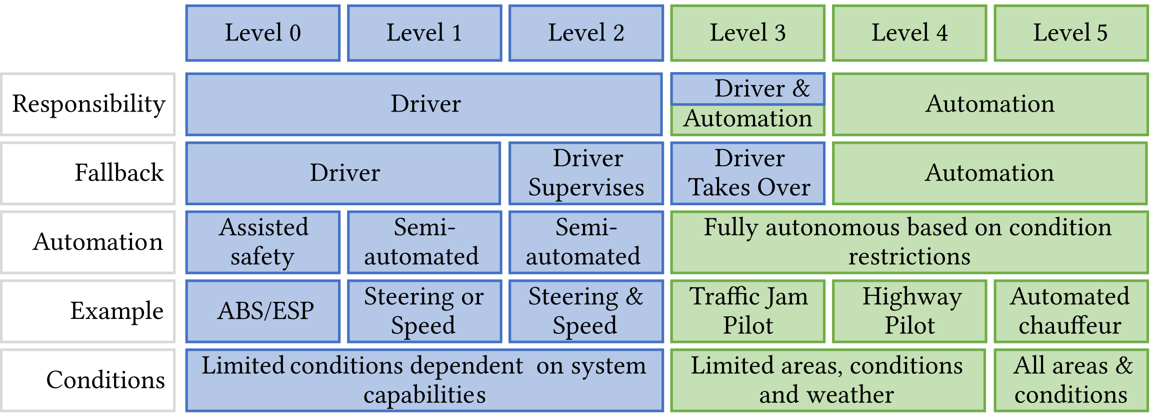
Figure 1.1: Overview of the proposed modular driver activity recognition system.

1.1 The Driver in Automated Vehicles

Studies show that the behavior of drivers as well as their responsibilities change depending on the automation capabilities of their car. At the time of writing fully autonomous cars are not commonly available yet. Current systems are classified on a scale of six automation levels which define both the capabilities of the automation system as well as the responsibilities of the driver (see Figure 1.2). These levels were introduced by the organization *SAE International*. They are commonly called *SAE Levels* [SAE21].

SAE Level 0 and *1* cars offer few automation functions. The driver remains in charge and must step in if necessary. Driver behavior and causes of crashes in these cars are well studied through accident report analyses [Ese22], natural driving studies [Din16] as well as driver interviews [Gor05]. While drivers drive safely most of the time, they are still a contributing factor to most accidents [Din16]. These factors can be categorized into four main groups: impairment (e.g., drowsiness, drugs), performance errors (e.g., overlooking a traffic sign), judgment errors (e.g., speeding) and distraction. Of these categories distractions contribute potentially to 36% of crashes in the United States [Din16] and 10-30% in the European Union [Com15]. The main distraction with high associated risk is currently handheld cell phone usage [Wor15, E-S19]. Other common activities include eating, drinking, reading, writing or reaching for an object [Gor05, Din16].

Automated vehicles were introduced commercially around the year 2015 [Eri17]. Their capabilities, even as of today, are mostly classified as *SAE Level 2*, relieving the driver fully from driving the vehicle. However, drivers are still responsible and are required to supervise the automation to take over if necessary. Because of their recent introduction, there are few real-world studies of driver behavior in these vehicles [Nor21]. To enforce supervision and to make it harder to bypass safety measures, drivers are required to keep their hands on the steering wheel. However, these measures are circumvented by some drivers [Lin18]. Because the driver is still responsible, even if not controlling the car directly, the findings for manual cars, regarding distractions and impairment, still apply. However, their impact can change, depending on whether the automation can handle the situation or not. In addition, both impairment and distractions are more likely because drivers are less occupied [Ban18, Nor21]. Even if drivers try to supervise their car constantly, they can get fatigued or distracted due to their low workload [Gre18]. On the other hand, an automation system that works well can lead to over-trust, where drivers stop supervising and start occupying themselves with other activities. This leads to an increase in frequency of the already presented activities, like mobile phone use [Nor21], or even new activities, like watching videos [Lin18]. Both low workloads and over-trust in the safety of these systems already led to accidents where the automation failed and the driver did not notice [Nat18].



	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
Responsibility	Driver			Driver & Automation	Automation	
Fallback	Driver		Driver Supervises	Driver Takes Over	Automation	
Automation	Assisted safety	Semi-automated	Semi-automated	Fully autonomous based on condition restrictions		
Example	ABS/ESP	Steering or Speed	Steering & Speed	Traffic Jam Pilot	Highway Pilot	Automated chauffeur
Conditions	Limited conditions dependent on system capabilities			Limited areas, conditions and weather		All areas & conditions

Figure 1.2: Overview of the *SAE Levels* for automated driving.

Automation systems classified as *SAE Level 3* and *4* are still in development. They further shift the responsibility from the driver to the automation system. The primary difference between these levels is the automation behavior on failure, which can occur because of technical problems or environmental influences. In these cases, *SAE Level 3* systems only guarantee safety for a short time and will return control to the driver. *SAE Level 4* systems, on the other hand, are capable of performing a minimal risk maneuver to ensure safety in all circumstances while they are active. These transition are challenging and are discussed separately in Section 1.2. In both modes the driver will be allowed to engage in other activities. It is still unclear how this will be handled for *SAE Level 3* vehicles because the driver must be able to react in a timely manner. For example, sleeping impairs the driver for too long [Hir20, Wör21]. It is expected that future vehicles will change automation levels depending on their current surroundings [Die22b]. For example, a car could have a highway pilot classified as *SAE Level 4*, where the driver is free to do other things, and could then change to *SAE Level 2* when exiting the highway. To the driver the difference is not clear because the experience is the same at first glance – the car still drives by itself. However, their responsibilities change. This, so-called "mode confusion" already exists with current *SAE Level 2* cars, where drivers assume the car is in automated mode but is in fact driving uncontrolled manually [Ban18, Wil20]. Mode awareness gets increasingly challenging with more possible automation states [Die22a].

SAE Level 5 systems are fully autonomous. They are required to work in all conditions a human driver can drive safely. They do not need direct control or intervention from the user and there is no dedicated driver. Passengers can do whatever they like. They do not need a driver's license and could even be impaired (e.g., drunk). However, there are still challenges for the automation regarding the interior. Currently, drivers are responsible for the state of the interior. They must check if passengers are seated properly and if items are secured. In the future this may also be a responsibility of the automation system.

1.2 Driver Monitoring for Takeover Scenarios

Until automated driving systems can handle all possible situations (*SAE Level 5*), the human driver serves as a fallback and has to take over the task of driving in a timely manner if a system boundary is reached. Takeovers can happen in *SAE Level 2* to *4*. However, in *SAE Level 2* the driver is always responsible and the automation can fail silently without warning the driver. For this reason, these automation systems disable themselves as soon as the user interacts with the controls. In higher automation levels the necessity for the driver to take over has to be detected by the automation system and can therefore be planned in advance. The driver is notified by a takeover request (TOR) if necessary. This section describes how drivers react in these scenarios. It also presents a concept of how driver monitoring can be integrated into the control loop of the automation. This concept is based on our publication in the journal *AT Automatisierung* [Lud18] © De Gruyter, 2018.

Determining the time span a driver needs to take over after a period of highly automated driving is subject to current research. Next to environmental factors, such as the criticality [Gol13] or the complexity [Dam12] of the traffic situation at the time of the TOR, the state of the driver is an important parameter for the takeover process. In higher automation levels (*SAE level 3-4*) the driver can pay attention to something or somebody else, which results in the problem that the capabilities of the driver after a TOR can vary significantly. In the following some empirical findings on the influence of the driver's state on the takeover process are presented:

Petermann-Stock et al. [Pet13] examined the influence of different levels of the driver's load on the time of a first driving action after a TOR. Therefore, they analyzed 38 non-driving related tasks which are likely executed during automated driving and designed three comparable tasks with low (cognitive and acoustic), mid (cognitive and visual) and high load (cognitive, visual and motoric). In an experiment with two TORs they found a significant influence of the load level on the reaction time with mean values of 2.0, 2.9 and 3.3 s and maximum values of 2.4, 8.8 and 8.3 s for the different tasks.

Radlmayr et al. [Rad14] created an experiment to investigate the effect of non-driving related tasks in varying traffic situations on the takeover process. The authors used the visually demanding Surrogate Reference Task (SuRT) and the cognitively demanding n-back Task as activities during automated driving. The SuRT group needed on average 2.16-2.71 s, the n-back group 1.93-2.92 s for their first driving action depending on the situation, which are a significant increase of the takeover time compared to the baseline group (1.55-2.32 s).

The participants of the study from Merat et al. [Mer12] had to supervise an automated vehicle with and without performing a secondary task, as well as drive manually with and without distraction. The secondary task was to guess items by asking a series of questions, answered with yes or no by the experimenter, creating a cognitive workload comparable to a telephone conversation. In absence of the secondary task the drivers reduced their speed in response to a critical incident independently of the automation level while distracted drivers did not. The authors suspect that distracted drivers are not as aware of their resource limitations and cannot adapt adequately.

The impaired driving performance after a takeover was also measured in other studies: Damböck et al. [Dam12] found a significant difference in the number of driving errors after a takeover request with time buffers of 4 and 6 s to the incident compared to their baseline group. Even with a takeover time of 8 s some participants executed the necessary lane change too late, not at all or in the wrong direction.

Gold et al. [Gol13] analyzed the reactions of distracted drivers to an obstacle on the lane with a TOR at a time-to-collision of 5 and 7 s, respectively. In comparison with the baseline group the distracted drivers executed stronger steering and breaking maneuvers with increased acceleration of two to three times. In addition, only 8 out of 26 lane changes were secured via a shoulder check, showing the high risk of a collision for this takeover scenario.

Merat et al. [Mer14] looked at the driving behavior for 60 s after the TOR and report that after a time interval of 10 s to regain control, the participants showed an increased number of steering corrections for the next 10-15 s, which then steadied after 35-40 s. As the visual attention is following the same trend, the authors state that 40 s is an adequate amount of time to resume control of driving from automation.

In summary, the presented studies show two effects: First, the cognitive state of the driver and his activity before takeover have a major influence on the takeover time and second, the driver is prone to driving errors after taking over from the automation. There are indications that even being physically able to take over control, drivers still lack awareness of the current situation and might not have mentally disengaged from their last activity. Present approaches for takeover management either only do a short fadeout of the assistance system after a takeover request [Pet13] or an entire shutdown after the first action of the driver [Mer12, Gol13, Rad14], which can potentially lead to unsafe and dangerous situations because of mistakes the driver makes after transferring control.

Based on these findings we propose a concept to guide the driver during takeover via driver monitoring and a haptic shared control scheme [Lud18]. This is a control concept where the input of the driver while taking over is moderated by the automation to ensure safety. The driver can feel the system correcting as resistive force on the controls. The correction force of the automation is gradually reduced over a period of time until the driver is fully in control. Within this control scheme a takeover consists of three parts. The point in time when the TOR is issued, followed by the time the driver needs to reach the controls and finally the time of shared control to conclude the takeover. While these intervals can be set to fixed values, there is potential for adaptation based on the presented studies, to increase both safety as well as comfort and acceptance of such a system.

We propose to adapt these times based on the state of the driver, determined by a driver monitoring system. While there are many other methods to determine focus of attention [Vor18], tiredness [Zha19a] or even takeover readiness [Lot19], we propose to apply the methods for body pose estimation as well as activity recognition presented in this thesis. This system provides the control algorithm with two time periods that we call *physical readiness time* and *mental readiness time*. *Physical readiness* describes the time the user needs to reach the controls and *mental readiness* describes the time the user needs to assess the situation and to disengage from his previous task. Both these time periods are processed by the control algorithm. They are used to optimize the time the TOR is issued and the period between *physical* and *mental readiness* is used to gradually hand over the control of the vehicle to the driver. We propose to determine these time frames as follows:

Estimation of physical readiness time is mainly dependent on the motoric capabilities and reaction time of the human body. While there are individual differences as well as other influencing factors, we are interested in the shortest time the driver needs to reach the controls, so we can ignore these influences because they make the reaction slower. This will inform the automation when to expect control input at the earliest. We propose to estimate *physical readiness* based on the distance between the hands and the controls of the car, determined by our activity detection system (see Section 5.1). The system should estimate this distance continually to update the time while the driver is getting ready.

Estimation of mental readiness is more difficult to assess because it is influenced among others by outside factors, like the complexity of the surrounding traffic, as well as by the mental state of the driver, which can only be observed indirectly. We propose to use the driver activity recognition methods contributed in this thesis to estimate *mental readiness*. In a first step we can use the findings on safe transition times for different secondary activities from the presented studies to parameterize the system. These parameters could then be further refined using the past transition performance of the driver. With this information the control system can estimate how long a transition will take overall. In addition, we can refine the estimate after the TOR by detecting if the driver gets ready, for example by putting away objects or reaching for the controls (see *physical readiness*).

The proposed concept is also supported by Lotz et al. [Lot19] who present a system for takeover time prediction of truck drivers considering similar time intervals for the transition and using driver body pose as well as eye gaze tracking as input features.

The development and realization of this approach took place within the publicly funded project PAKoS (personalized adaptive cooperative systems for automated vehicles)[1]. The concept was developed and published in equal parts together with Julian Ludwig. Manuel Martin focused on the concept for driver monitoring while Julian Ludwig focused on the control algorithm. While it was not possible to realize the whole concept within the project, Julian Ludwig showed the effectiveness of the control scheme in driving simulator studies [Lud21]. This thesis contributes the necessary driver monitoring systems including driver body pose estimation as well as fine-grained driver activity recognition.

1.3 Applications for Driver Monitoring

As already elaborated, a key requirement for driver monitoring is to detect the driver's state to inform the automation about the capabilities of the driver to either monitor the automation (*SAE Level 2*) or to take over (*SAE Level 3-4*). These requirements are also recognized by the legislature of more and more countries, like the European Union [Eur19] or USA [DeF20], making driver monitoring system mandatory in the near future for all vehicles with automation functions.

Apart from the safety aspects discussed in previous sections there are additional applications for driver monitoring regarding comfort functions. In the following paragraphs we will summarize key applications for driver monitoring where the systems presented in this thesis can contribute the most:

Distraction detection is a well outlined application based on the previous sections and legal requirements. It will be essential for all future vehicles that are not fully autonomous to prevent accidents in general as well as to inform the automation about the driver's ability to take over. While current preproduction systems mostly rely on head pose estimation, eye tracking and facial features to infer an abstract distraction or tiredness score [Man20], we think that activity detection will be valuable for systems with a higher automation level to further improve the distraction detection score and to identify the source of distractions.

Take over management is an extension of distraction detection. Driver monitoring systems need to at least detect if the driver is able to take over. However, we think that future systems will also use driver monitoring to estimate how long a takeover will take or if the driver is getting ready after a takeover request. We describe this concept in Section 1.2.

[1] https://www.interaktive-technologien.de/projekte/pakos, accessed: June 15, 2022

Level appropriate behavior management addresses the driver's different responsibilities depending on the current automation mode. However, it is challenging in that there are no major differences for the driver in different automation levels as long as the car drives by itself without issue. This can lead to mode confusion or abuse where the driver does not behave as they are expected to (e.g., monitoring in *SAE Level 2*, being ready to take over in *SAE Level 3*). Activity recognition can help to solve this challenge by warning and explaining the issue if the driver's actions are not appropriate. For example, when the driver is performing an inappropriate activity while in *SAE Level 2* mode they could be informed that they are not allowed to do so currently. In addition, if the system can identify the activity it could recommend to continue the activity at a later time if there is a section with a higher automation level on the planned route. This would both train the driver, explain the system's reaction, and offer a comfortable solution. We investigate this concept in our current publicly funded project KARLI[1] based on the algorithms developed in this thesis.

Motion sickness prevention is another application for driver monitoring. Many passengers get sick in cars, especially if they occupy themselves with other activities without observing the road for a prolonged time [Sch20]. While this posed less of a problem for drivers in manually driven vehicles, it can be a challenge in higher automation modes. This can also have safety implications if the driver needs to take over while being impaired. The application of driver monitoring using, among others, activity detection methods contributed by our work, together with a recommendation system to prevent motion sickness is another avenue explored in project KARLI.

Personalized interior adaptation is another application for future driver monitoring systems. Based on 3D human body pose estimation it is possible to configure the seat, the mirrors as well as the steering wheel when entering a vehicle. This would be especially useful in situations where the driver changes often, like in rental cars or with car sharing services.

Novel User Interfaces are a continuously researched topic by automotive manufacturers to improve the driving experience. There are already infotainment systems reacting to the driver when reaching for the interface, using proximity sensors, as well as gesture recognition systems to control, for example, the audio system of the vehicle. Both applications are still restricted to small areas within the car and require special sensors for each function [Raj18]. The camera system as well as algorithms proposed in this thesis can also be used for these applications with the option to expand the acquisition area substantially.

[1] https://karli-projekt.de, accessed: June 15, 2022

Digital personal assistants are another topic where driver monitoring can help. While speech recognition is already widely in use in vehicles, the performance of the technology was strongly improved with the introduction of smart speakers for home use [Pur17] and their introduction for automotive applications. Currently, these systems only react to the user when asked using a "wake word". In the future these systems could use interior monitoring capabilities to make the wake word obsolete. In addition, they could provide proactive recommendations [Wei22], for example regarding breaks for gas or getting something to eat or drink based on activities of the passengers inside the vehicle.

1.4 Contributions and Outline

This thesis focuses on driver activity recognition for future automated cars with a wide variety of applications as presented in the previous sections. One of the challenges encountered in this thesis is the scarcity of data to reach this goal. Our first contributions are therefore multiple datasets to evaluate parts of our system leading to currently one of the biggest public datasets for driver activity recognition for automated cars. We propose a modular driver monitoring system. Our approach consists of two stages where only the first stage is dependent on the used sensors while the second stage, estimating the driver's activities, stays sensor independent. This approach allows us to combine datasets from diverse sources including data from other domains for parts of the system. In addition, it allows to reuse labeled data for activity recognition even with sensor modality and viewpoint changes. We demonstrate these capabilities in Section 5.4.4.

To achieve this goal, we propose an abstraction layer using 3D data followed by driver activity recognition using this abstraction. The main feature in this layer is the 3D driver body pose. Its estimation is in itself a difficult problem where we contribute two real-time capable approaches. In addition, we enrich the abstraction layer with a model of the interior of the car as well as with 3D tracks of objects used for certain activities (e.g., water bottles, food) to form a complete 3D interior state model. Based on this 3D state model we perform driver activity recognition. Our main contribution for activity recognition are approaches that can integrate additional features, like the interior model or objects positions, in addition to the 3D driver body pose. This is also a novelty for 3D body pose based activity recognition systems in general.

To our knowledge the proposed system is currently the only approach that tackles driver activity recognition using 3D data from the ground up. An overview of the method is depicted in Figure 1.1. The structure of the remainder of this thesis with its main contributions is as follows:

Chapter 2: Related Work

This chapter presents an overview of the related research topics regarding datasets, human body pose estimation as well as body pose based activity recognition. The methods proposed in this thesis rely heavily on techniques from other non-automotive research areas. For this reason for each topic we depict the state of the art in the general case as well as in the automotive domain. The chapter finishes with a discussion of the identified research gaps that are addressed in this thesis.

Chapter 3: Datasets

Both driver body pose estimation as well as activity recognition rely heavily on machine learning and therefore need suitable datasets for development and testing. Especially with regards to depth data, there were no suitable public datasets for our intended plan. We therefore first analyze suitable sensor systems and sensor positions via literature research as well as simulation. On this basis we collect three datasets. Their main contributions are:

- The first dataset focuses on 3D driver body pose estimation based on depth data. It uses an automotive-certified prototype depth camera and is manually labeled. The dataset covers the most common movements while driving manually and includes some more challenging poses regarding smartphone usage and self-occlusion (e.g., crossed arms, reaching for the footwell). While the data is not publicly available it enabled our research on this topic [Mar17a]. (Section 3.3)
- We also collect the first dataset for driver activity recognition that includes depth data, and video data with high frame rate. It enabled our research on 3D driver body pose-based activity recognition. It is also the first to include a model of the interior of the car [Mar18b]. (Section 3.4)
- The final dataset expands on our previous experiences. Its focus are complex driver activities for automated vehicles. It goes beyond the scope of the previous datasets in all areas, using a multi-view camera system, hierarchical labeling for fine-grained driver activities, an interior model as well as pre-computed 3D driver body pose labels. The dataset was published at ICCV 2019 under the name Drive&Act [Mar19]. We extended the labels of the dataset with additional bounding box and 3D-position data of objects used while performing different activities [Mar20a]. In addition, we use Drive&Act data to construct and publish a challenging benchmark for 3D driver body pose estimation with varied scenes of different activities including objects and occlusions [Mar21]. (Section 3.5)

Chapter 4: 3D Driver Body Pose Estimation

The focus of our work on 3D driver body pose estimation is real time capable approaches based on data from depth cameras. One of the challenges in this area is the scarcity of training data. We contribute two methods that deal with these challenges in different ways:

- Our first approach uses simulated depth data to train a system using decision forests. The method achieves real time performance even on CPUs which was not possible at that time using methods based on convolutional neural networks. To our knowledge this is also the first published method for 3D driver body pose estimation with a thorough evaluation of its accuracy [Mar16, Mar17a]. We demonstrate its ability to generalize to real data by evaluating it on both real-world datasets collected in this thesis. (Section 4.2)
- The main contribution of our second approach is the decomposition of the 3D body pose estimation task into 2D body pose estimation on video images followed by 3D body pose estimation via the 2D body pose and depth images. With this approach the first part can rely on any state of the art 2D body pose detector, making use of public large scale image datasets for training. Only the second part relies on limited amounts of annotated depth data from the car interior. The method uses a deep neural net to infer the 3D pose based on the input. Our evaluation also shows the robustness of the method to occlusions which is a challenge for depth image-based 3D driver body pose methods [Mar21]. (Section 4.3)

Chapter 5: Driver Activity Recognition

The driver activity recognition methods proposed in this thesis built upon each other further increasing the complexity of the input as well as the flexibility with regards to input changes. All our methods investigate the premise that the distance of the driver to objects or interior elements is relevant for activity recognition methods. Our main contributions are:

- We first demonstrate the advantages of 3D driver pose estimation combined with a 3D model of the interior to detect interaction with interior elements (e.g., hands on wheel detection, grabbing the gear lever). Compared to similar approaches our method is simpler, requires fewer computing resources, and can easily be adapted to other cars or additional areas of interest [Mar17a]. (Section 5.1)
- Expanding on this technique we combine interior elements with the driver body pose in a multi stream recurrent neural network to determine the driver activity.

We show that the addition of the interior model helps improve the activity detection performance especially for activities where the location is relevant for the action [Mar18b]. (Section 5.2)

- Finally we propose a method that also integrates the location of objects. We model the input representation consisting of 3D driver body pose, interior elements and object locations as a graph. Our main contribution is the graph creation process that works on the principle that objects close by are more relevant for activity recognition than objects further away. We use a graph convolution-based method to process the input and can show the effectiveness of our graph creation method as well as the advantages of adding information about objects in the scene [Mar20a]. (Section 5.3)
- Lastly, we test the graph convolution-based method for robustness to sensor and viewpoint changes as well as for cross dataset performance and can show its effectiveness compared to end-to-end video-based methods. (Section 5.4.4)

Chapter 6: Conclusion and Outlook

We summarize our results and place them in the context of the related work both for computer vision in general as well as regarding automotive applications. In addition, we discuss future improvements of our approach, new research within the automotive context and also how our contributions can be applied to other domains.

2 Related Work

Our research on driver body pose estimation as well as driver activity recognition was based on a large body of prior work. In many cases methods for driver monitoring were themselves based on advances in the general area of research. In this chapter we therefore present an overview of the state of the art for body pose estimation as well as activity recognition in general, followed by an in-depth review of automotive methods.

2.1 General Human Body Pose Estimation

Human body pose estimation is a challenging computer vision problem because of the high degrees of freedom of the human body. It involves the estimation of the location of a set of keypoints (e.g., wrists, shoulders) of the human body. To be able to develop these methods and to ensure their generalization capabilities to real world applications, large datasets are necessary. In the following we distinguish between 2D and 3D human body pose estimation and present the current state of the art both for datasets as well as methods. We focus mostly on approaches for depth image-based 3D human body pose estimation, because it is the main focus of our own work, while also giving an overview of other approaches for this task.

2.1.1 Datasets

Collecting datasets for 2D as well as 3D human body pose estimation poses different challenges. Datasets are therefore usually specialized for one of these tasks. In the following we present an overview of current state of the art general datasets as well as the challenges when collecting and annotating the data. This section serves as a comparison for driver body pose estimation datasets (see Section 2.2.1).

2D human body pose estimation datasets require pixel-accurate annotations of keypoints of the human body. These datasets are usually annotated manually by annotators marking the position and type of each keypoint for each person of each image of the dataset. The annotation process is challenging because lighting and shadow can cause visibility problems. In addition, depending on the body shape or clothing, the position of keypoints cannot be determined precisely. Keypoints are also often occluded either by the environment or by self-occlusion. If body parts are only partially occluded the position of the keypoint may still be estimated by the annotator. Some datasets mark these landmarks as occluded. In case of severe occlusion annotation is not possible. For automotive datasets for interior monitoring, occlusions are a frequent problem because of the confined space.

There were multiple smaller datasets for methods predating the now popular deep learning-based approaches. They all consisted of 10 000 images or less [Joh11, Sap13]. They were used to develop the first deep learning-based approaches, however, their size was limiting for more complex methods.

Andriluka et al. [And14] identified this problem and published the *MPII Human Pose* dataset consisting of about 40 000 images collected from YouTube videos. The dataset was not only much larger than previous datasets, it also offered more variety in all aspects (e.g., background, body orientation and human body pose).

The current state of the art for 2D human body pose estimation datasets is the *Microsoft Common Objects in Context (MS COCO)* dataset [Lin14]. It is a large dataset consisting of 330 000 images with multiple annotations, amongst others, object bounding boxes, pixel level masks for each object as well as 2D human body poses. The images were collected from different online sources with a focus on non-iconic images. Which means images showing the main class in context with other classes, for example a person riding a bike instead of a portrait of a person. The authors argued that this helps with generalizing and preparing for novel environments that have yet to be encountered.

Methods trained on the data are often used as building blocks for further tasks, like action recognition, as we will show in the next sections. We also used methods trained on this dataset as part of our approach for 3D driver body pose estimation (see Section 4.3). While methods trained on this dataset were often used in automotive applications, their performance on automotive data was usually not evaluated quantitatively. We contributed such an evaluation as part of our work (see Section 4.4.1).

3D human body pose estimation datasets are difficult to generate with a large variety of people as well as different backgrounds because it is not possible to accurately annotate 3D locations of keypoints on images from a single point of view. Collecting image or video data with high variance from the web is therefore usually not an option. Instead, data is often collected in dedicated experiments with specialized hardware for annotation purposes. In the following paragraphs we discuss the most common methods to annotate general 3D body pose datasets. For a detailed summary of current state of the art datasets for 3D human body pose estimation we refer to Wang et al. [Wan21a].

Compared to the 2D human body pose estimation task 3D coordinates cannot be annotated in a single color or near-infrared image because the depth information is lost when projecting the scene onto the sensor. If there are multiple cameras in a calibrated multi-view system 3D data can be reconstructed via triangulation. In this case manual annotation of 2D keypoints on all views followed by triangulation is sufficient. However, the drawback of this approach is its expense because it is necessary to manually annotate multiple images to retrieve a single 3D human body pose. The resulting datasets are therefore generally small. However, compared to the annotation methods, presented in the following, there are no additional sensors just for annotation. This keeps the complexity of the recording setup low and enables recording data flexibly, even outdoors [Bel14].

Optical Motion capture systems are designed to capture the movement of a person with high spatial as well as temporal accuracy. They are used in the movie and game industry to capture the performance of actors for animation purposes. These systems are also a popular choice for collecting datasets for 3D human body pose estimation. For this purpose, they are usually combined with additional cameras that provide the image data while the motion capture system provides the annotation. Marker-based systems rely on markers fixed to the recorded person at known locations. The motion capture system then locates these markers to determine the 3D human body pose with high accuracy. These systems often work with active near-infrared (NIR) light sources. They are therefore limited to indoor environments because sunlight interferes with their operating principle. In addition, the markers are visible in the collected data and restrict the choice of clothing to limit the occlusion of markers [Sig10, Ion14]. There are also marker-less motion capture systems which offer more flexibility with regards to the environment and clothing [Joo16, Meh17a]. The main drawback of this annotation approach is the limited recording area that does not allow for much variation of the background. This is a result of the high complexity of the system, requiring many motion capture cameras all around the recorded space to limit the occlusion of markers.

Depth cameras offer another solution to retrieve 3D data with a single camera. Datasets including depth images can either be annotated using independent motion capture systems [Gan10, Gan12, Ofl13], as presented above, or they can be annotated using the depth data itself with manual annotation [Sri21] as well as automated depth image-based approaches [Haq16, Wan16a]. However, relying on just depth data from a single point of view for annotation limits the quality of the annotation with regards to occlusions. We analyze the challenges of using depth images for 3D human body pose estimation in detail in Section 4.1.

There are also motion capture systems that rely on inertial measurement units (IMUs). However, their results are usually relative to a calibrated starting position. In addition, these systems experience sensor drift which reduces their reliability over time.

These drawbacks make it challenging to calibrate IMU-based motion capture systems to cameras for 3D annotation [vMar18].

Because of the discussed complexities when recording such datasets, synthetic data is an interesting alternative to generate depth [Sha16a, Mar18c] or color [Var17] images with precise automated annotations. However, the generalization to real world data is challenging for methods trained on synthetic data because of the remaining differences between simulated and real images.

For our work on driver body pose estimation we relied on synthetic data (see Section 4.2.2) as well as triangulated data (see Section 4.3.2) for training. For testing we used both manually annotated depth images (see Section 3.3) as well as manually annotated and triangulated multi-view data (see Section 3.5.4).

2.1.2 2D Human Body Pose Estimation

With the advances in deep learning-based image classification [Kri12] and the introduction of large datasets [And14, Lin14] the first methods for 2D human body pose estimation based on convolutional neural networks (CNNs) were introduced. They focused on body pose estimation of single persons using small neural networks applied to image patches with a sliding window approach [Tos14, Tom14, Jai15a]. This approach enabled training on small parts of images, limiting the required computational resources. Later fully convolutional models became the norm using network structures for multi-scale feature extraction [New16, He16, Wei16, Che18].

Modeling the output of neural networks for efficient human body pose inference was an important part of this research area. There were two basic approaches: Regression of body keypoint coordinates [Tos14] or classification of body keypoint locations via heatmaps [Jai15a, Wei16, New16, Sun19]. The latter method became the most popular approach because regression-based methods proved to be difficult to train and less accurate [Tom14, Sun18]. However, this created additional challenges for approaches pushing the accuracy boundaries because of the typically low resolution of heatmaps. This led to methods combining heatmaps for coarse localization and regression to compensate for the discretization caused by the heatmaps [Pap17, Kre19].

While earlier deep learning-based methods focused on single person pose estimation, handling multiple people in the image is necessary for many real world applications. This requires to determine which part of the image and therefore which body part belongs to which person. The two main approaches to solve this problem were a top-down or buttom-up analysis of the image:

Top-down methods apply a person detector to crop relevant areas for single person pose estimation. With this method any single person pose detector can be applied to multiple people. This approach works well for few people in the image because it ensures high resolution for body pose estimation via cropping and simplifies the task to single person pose detection. However, crowded scenes are challenging because partially occluded people might not be detected and the cropped image can still contain parts of multiple people making the body pose estimation task ambiguous [Fan17]. In addition, the processing time increases linearly with the number of people in the image because the pose detector has to run for each detected bounding box. He et al. [He17] proposed one of the first end-to-end trainable approaches for both person detection, including instance masks, as well as human body pose estimation via heatmaps. Fang et al. [Fan17] proposed a method to handle bounding boxes with body parts from multiple people by training a network to deform the image to focus the pose detector on the primary target. Papandreou et al. [Pap17] increased the accuracy further by regressing offsets in addition to heatmaps to compensate the discretization error of low resolution heatmaps. Bazarevsky et al. [Baz20] focused on real time performance on mobile phones with reduced accuracy by directly regressing the human body pose in the bounding box.

Bottom-up methods first detect all keypoints of the people in the image followed by a method to group keypoints to the full body pose of each person. They analyze the whole image at once and can therefore handle crowds well because they are not constrained to isolated individuals within bounding boxes. Their speed is therefore also largely independent of the number of people in the image. However, because they handle the whole image at once, the resolution per person within the detection system is lower, which can decrease the localization accuracy. Most of these approaches use heatmaps to determine the position of each keypoint. However, with multiple people in the image this leads to multiple peaks in each heatmap. This makes it necessary to identify which peak belongs to which person. This is often approached by building a graph using unary weights, which is typically the height of the peak in the heatmap, as well as binary weights to model the probability that keypoints belong together. Solving the resulting graph matching problem leads to an efficient grouping of keypoints to complete body poses.

While this problem could be solved using keypoint heatmaps together with geometric models [Pis16], a popular approach was to regress additional features as binary weights within the neural network. In most cases the regressed binary features represented some form of vector field that indicated the pairwise connection between keypoints by pointing from one to the other. There were different approaches to generate these vector fields like regressing vector offsets from each keypoint to each other keypoint of the same person [Ins16] or to neighboring keypoints in the kinematic chain

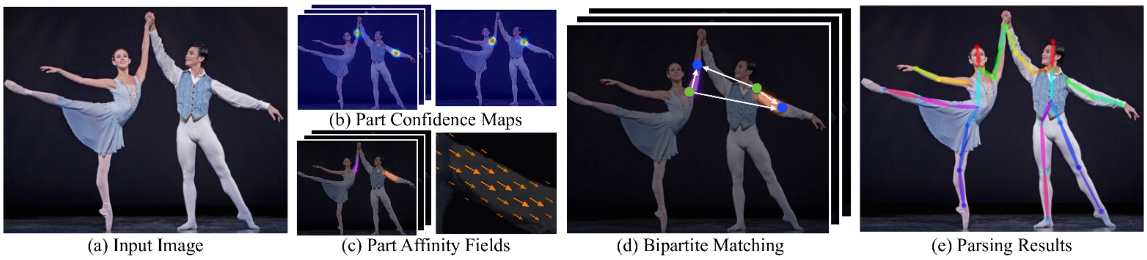


Figure 2.1: Overview of the OpenPose method for 2D human body pose estimation [Cao18].

of the human body [Pap18, Kre19] or direction vectors pointing in the direction of neighboring keypoints [Cao17]. Like top-down methods later approaches regressed short range offsets in addition to heatmaps [Pap18, Kre19] as well as offsets to both ends of keypoint pairs as binary features [Kre19]. This made the accuracy of these methods largely independent of the resolution of both heatmaps and vector fields.

OpenPose [Cao17, Cao18] is a 2D human body pose estimator that is popular for activity recognition systems, especially in the automotive domain. It creates both heatmaps as well vector fields, called affinity fields, that point from one keypoint to the next belonging to the same person. Figure 2.1 shows an overview of the approach. First local maxima are extracted from the heatmaps then part affinity fields are sampled between all pairs of neighboring keypoints in the kinematic body model to generate keypoint pair weights. This data is then cast into a graph and the graph matching problem is solved efficiently with a greedy heuristic approach resulting in the 2D body pose of all people in the image.

We relied on OpenPose to generate 3D driver body pose annotations for the Drive&Act dataset via triangulation (see Section 3.5.4) as well as to estimate the 2D body pose as a starting point for our depth image-based 3D driver body pose estimation approach (see Section 4.3).

2.1.3 3D Human Body Pose Estimation from Depth Images

While methods for 2D human body pose estimation improved by a large margin, thanks to large benchmarking datasets, data was less available for depth image-based 3D human pose estimation methods. Therefore, adoption of machine learning methods took longer, especially with regards to data-dependent deep learning-based approaches. In addition, methods were not easily comparable because they did not use common datasets or even used private data sources. On the other hand, depth data is well suited for simulation using computer graphics because lighting and texture are not relevant. This enabled the creation of large well-labeled training datasets [Sho11, Buy14, Hes15, Raf15]. Analytic methods, often 3D model and tracking-based approaches, were popular because the segmentation of people in depth images worked well for many applications and 3D data was well suited to fit volumetric models [Gan12, Baa13, Hel13, Din15].

The research area changed rapidly with the introduction of the Kinect as affordable depth sensor, and the related papers using Random Decision Forests for 3D human body pose estimation. In the following we give an overview of data driven approaches using Random Decision Forests followed by state of the art methods based on deep learning.

Random Decision Forests for body part labeling became popular with the approach by Shotton et al. [Sho11]. In general, most of these approaches relied on a segmentation of each person from the depth image. This was often achieved by analytic approaches and heuristics using background segmentation. Each segmented person was then processed separately. Shotton et al. [Sho11] classified each point of the segmentation with a Random Decision Forest to assign body part labels. They then applied mean-shift clustering to infer the final keypoint locations in 3D using the intrinsics of the camera to re-project the depth image into the 3D scene. To train the Random Decision Forest they required data of people in as many different poses as possible with accurate body part label annotation. They therefore rendered synthetic depth images with matching annotation using 3D models of humans with different body proportions and a large database of motion capture data. This method was also the basis of the Microsoft Kinect SDK, providing accurate 3D human body tracking at low cost [Wan15]. The results of the SDK were popular for activity recognition systems and datasets [Sha16b, Liu19a].

We proposed a similar approach for our first driver body pose estimation system because of the low resource requirements of Random Decision Forests and the good quality of the resulting 3D driver body pose (see Section 4.2).

Girshick et al. [Gir11] improved this method using Random Decision Forests to regress offset vectors to keypoint positions. Different parts of the approach were further optimized speeding up the approach using Random Decision Ferns [Hes15] or Random Tree Walks [Yub15] or increasing accuracy by combining classification and regression [Sho13] as well as by regressing parameters of articulated body models [Tay12]. Others developed multi-stage methods, first regressing the global rotation of a person then applying a Random Decision Forest specialized for the respective view [Lal14] or using the Random Decision Forest to both create the segmentation of the person as well as to classify this segmentation for body parts [Buy14]. Rafi et al. [Raf15] handled occlusion explicitly by generating simulated training data including common household items as foreground objects and by introducing an extra label to mark occluded regions of the segmentation. Jung et al. [Jun16] decomposed the task into keypoint localization followed by keypoint identification using a learned dictionary of body poses.

Deep Learning-based methods first followed the same approach as Random Decision Forest-based methods by densely labeling body parts of a segmented depth images. However, methods based on advances in 2D human body pose estimation using heatmaps as well as regression-based methods proved to be more successful.

Jiu et al. [Jiu14] replaced Random Decision Forests with shallow convolutional neural networks for body part labeling on depth image patches. Shafaei et al. [Sha16a] used a neural network for semantic segmentation to process the whole depth image and to label all body parts at once. This allowed the network to consider the global context instead of local patches.

Wang et al. [Wan16a] adopted a similar approach to single person 2D human pose estimation by predicting heatmaps for each body part but instead of color images they used depth images as input. Martínez-González et al. [Mar18c] expanded this idea to multi-person pose estimation predicting both heatmaps as well as part affinity fields [Cao17] to group body parts to persons. They trained on synthetic data using real depth images as background. Moon et al. [Moo18] converted the depth image to a voxel-grid and performed 3D human body pose estimation with 3D data from the start by using 3D convolutions and estimating 3D keypoint heatmaps.

Regression-based methods for 3D pose estimation with depth images were often combined with skeleton-based priors. Haque et al. [Haq16] determined the mean body pose on the training set of their data and used recurrent neural network to regress correcting vectors to adapt the initial mean pose to the image data in multiple iterations. Marín-Jiménez et al. [Mar18a] computed a set of reference body poses by clustering the training set and regressed weights for each reference pose via a neural network to infer the final body pose from the input image via linear combination.

All methods presented in this section up to this point relied solely on depth data. However, with the introduction of large-scale datasets for 2D human body pose estimation based on color data, it was advantageous to combine them with the more limited datasets including depth images. Zimmermann et al. [Zim18] achieved this by combining the 2D heatmap result of OpenPose [Cao18] with a voxel-grid representation of the depth image followed by a 3D convolution-based network to determine 3D heatmaps. Martínez-González et al. [Mar20b] decoupled depth prediction and 2D human body pose estimation. They first predicted the 2D body pose using a part affinity-based multi-person pose detector [Cao17]. They re-projected the resulting 2D human body keypoints to 3D using the corresponding depth image and used this initial 3D body pose as input to regress refinement vectors with a deep feed forward network.

We used a similar decomposition method to combine large-scale 2D human pose estimation datasets with our limited depth data from the car's interior (see Section 4.3).

2.1.4 Other 3D Human Body Pose Estimation Methods

Our main contributions to 3D driver body pose estimation were depth image-based. However, we also developed an approach based on the triangulation of 2D human body poses from multiple-views. This method served to annotate the Drive&Act dataset (see Section 3.5.4) and as a baseline for our depth image-based contributions to 3D driver body pose estimation (see Section 4.3.2). In addition, our methods for 3D driver body pose estimation were inspired by approaches for monocular 3D body pose estimation (see Section 4.3). In the following we provide a brief overview of related methods in both areas. For an in-depth review we refer to Wang et al. [Wan21a].

Triangulation-based 3D human body pose estimation relies on a calibrated multi-view camera system with two or more cameras. With an increasing number of redundant views both the accuracy and robustness to occlusions increases but the necessary computing power rises as well.

A good baseline for 3D human body pose estimation via multi-view systems was 2D body pose estimation followed by separate triangulation of each keypoint [Don19]. The results could be further improved by fitting a 3D pictorial structures model to the 2D observations [Don19]. In the case of multiple people the task gets more complex because it is necessary to find the right association between detections in multiple views for a successful 3D reconstruction. This could be done for example with a neural network for person re-identification [Don19] or with geometric constraints [Kad20]. It was also possible to extend 3D pictorial structure models for multi-person 3D body pose estimation [Joo16]. There were also end-to-end trained methods, using neural networks, inferring the 3D body pose of multiple people directly from the images of a multi-view system [Tu20]. However, while this approach achieved high accuracy it needed a large training dataset in the target domain.

Our triangulation-based approach relied on separate triangulation per keypoint to generate 3D body pose ground truth as well as high quality 3D body pose annotations of the Drive&Act dataset. While there were methods with higher accuracy, they required training data in the target domain which was not available at scale for 3D driver body pose estimation.

Monocular 3D human body pose estimation methods require just a color image or a video sequence from one viewpoint as input. This is an ill-posed problem because the depth information is lost when projecting the scene onto the image plane. The only way to reconstruct the depth data relies on image cues. It is therefore also difficult to determine the absolute 3D positions of body parts or the size of the person. Monocular methods were therefore often limited to estimate the 3D human body pose in a pose local and sometime scale invariant coordinate system. Deep learning-based methods for this task can be categorized into image or video-based approaches and lifting methods using one or more 2D human body poses as input.

Image-based methods require large datasets of image data with 3D body pose annotations. These datasets were usually recorded in laboratories using motion capture systems for automated labeling (see Section 2.1.1). Methods trained on just these datasets did not generalize well to real world applications because the people, clothing as well as surroundings were not varied enough. A common approach was therefore multi-task learning combining real world datasets for 2D human body pose estimation with less varied 3D annotated data [Xia17]. The output of these networks could be regression-based, determining the 3D position of each keypoint relative to the root keypoint of the body [Li15]. However, regressing the parameters of parametric body models produced better results because the constraints of the model simplified the regression task [Kan18]. Instead of regressing the 3D keypoint positions it was also possible to determine heatmaps for each coordinate axis [Meh17b].

Lifting-based methods do not require image data [Wan21a]. They can therefore be trained on any motion capture dataset and can still generalize to real world data. However, there are ambiguities where one 2D human body pose can be lifted to different valid 3D body poses. While image-based methods can resolve these cases using image cues, lifting-based methods cannot. Martinez et al. [Mar17b] proposed an effective baseline method lifting a single 2D human body pose to 3D via a small feed forward network. Zhao et al. [Zha19c] reduced the impact of ambiguities by integrating image features, extracted from a 2D human body pose detector, and by enforcing temporal consistency between time steps with recurrent neural networks.

Our depth image-based 3D driver body pose method (see Section 4.3) was inspired by the lifting approach of Martinez et al. [Mar17b]. However, instead of 2D human body poses we already used 3D human body poses as input. Our approach therefore did not suffer from the ambiguity problems of the lifting process and could regress absolute positions for each keypoint. Its main purpose was to fix errors of the input introduced by depth images in combination with body part occlusions (see Section 4.1).

2.2 Driver Body Pose Estimation

Driver body pose estimation is human body pose estimation for the interior of cars with a focus on the driver. These methods usually focused on the upper body of the driver down to the hips or knees because the legs are often occluded by the steering wheel or foot-well.

We conducted a literature review of the field by searching for driver body pose related keywords as well as by aggregating all papers that cited public datasets for this task. We filtered the resulting list for contributions that either published a dataset for driver

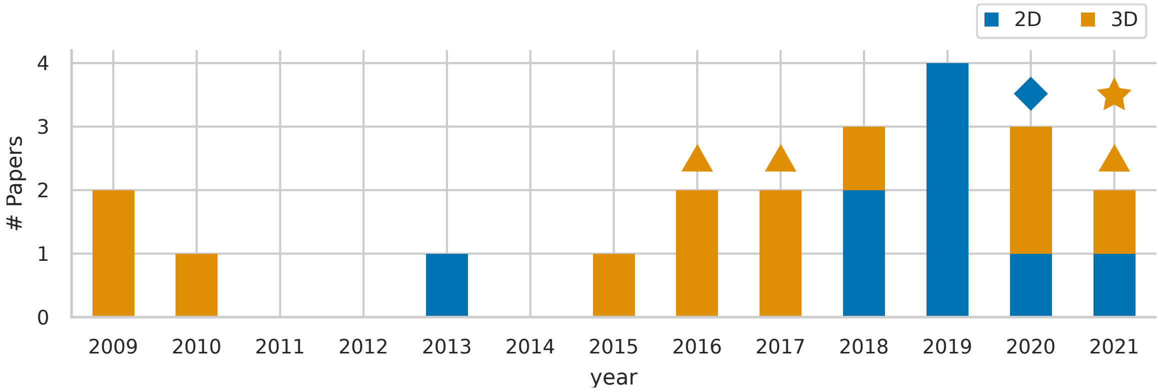


Figure 2.2: Number of papers per year for 2D- and 3D driver body pose estimation. ▲ indicates own publications. ★ indicates own public datasets. ◆ indicates the first public dataset.

body pose estimation or contributed an approach for the task. Overall, we discovered 21 relevant papers from 2009 until the end of 2021 (see Figure 2.2). To the best of our knowledge this represents the main body of work on driver body pose estimation in the last decade. Compared to general human body pose estimation the research area with an automotive focus was small. Research interest increased since 2015 but the introduction of the first public benchmarking datasets took until 2020. Compared to general datasets, automotive datasets for this task remain small, making it challenging to train methods just on automotive data. In the following we present our findings for datasets as well as methods in detail.

2.2.1 Datasets

Compared to the number of datasets for general human body pose estimation there were few datasets for driver monitoring which were in general much smaller. While public datasets were common in the general setting, data usually remained private for automotive applications [Mar17a, Mur17, Chu19, Yue19]. This started to change in recent years [Das15, Bor20, Dia20, Gue21, Mar21]. Table 2.1 shows an overview of, to our knowledge, all public and some private datasets for driver body pose estimation in comparison to general datasets. Compared to those datasets the use of near-infrared cameras was much more common in the automotive context. The reason for this is the requirement for driver monitoring systems to work at night and in daylight. This necessitates an active illumination system in the near-infrared spectrum combined with a camera that is sensitive for these wavelengths. We discuss the requirements on the sensor system in Section 4.1.

Das et al. [Das15] published one of the first datasets for hand detection within the vehicle showing a wider view of the interior of the cabin. However, the dataset was limited to bounding boxes of the hands of the driver and co-driver. In the following years work on driver body pose estimation was conducted only on private datasets including our first contribution in that area on 3D upper body pose estimation from depth images [Mar17a]. Our dataset consisted of 20 000 depth images recorded in a driving simulator. Test participants were instructed to perform a sequence of common movements while driving a vehicle. Keypoints were manually annotated (see Section 3.3). Others focused on 2D driver body pose estimation detecting either upper body keypoints [Chu19] or just elbows and hands [Yue19]. Both used private datasets recorded in normal traffic with manual annotation.

Borges et al. [Bor20] published the first benchmark dataset for driver body pose estimation. They investigated how to annotate 3D human body pose data accurately and automatically within the vehicle. As depicted in Section 2.1.1 the most accurate approach would be marker-based motion capture. However, because of the confined space of the car interior positioning motion capture cameras is difficult and occlusions cannot be prevented. Their solution was therefore to combine an optical marker-based system with inertial measurement units (IMUs) that do not require visibility of the body parts. They used this approach to annotate data of a depth camera. Overall, the dataset consisted of 8700 depth images collected while driving manually in traffic. Feld et al. [Fel21] described a similar setup using just marker-based motion capture in a more spacious driving simulator but without publishing data.

Dias Da Cruz et al. [Dia20] approached the annotation challenges in a different way by simulating data of the interior of the car. This enabled the export of high quality annotations not only of the body pose of the passengers but also other features that are time-consuming to annotate (e.g., labels for instance segmentation). They also generated data from different vehicles by changing 3D models and textures. Their published dataset consisted of 25 000 rendered depth, color and near-infrared images of 10 vehicles with, among others, 2D keypoints of the whole body of each passenger.

Guesdon et al. [Gue21] collected a large video dataset of people driving a car in normal traffic. The data included challenging lighting conditions that occur while driving in daylight. They selected 10 000 images from the videos based on image similarity and illumination differences and annotated the 2D driver body pose manually for each image.

Table 2.1: Summary of datasets for body pose estimation. 2D- and 3D general datasets are provided as reference. The table of public automotive datasets is a complete list for the last decade. Our own contributions are shown in bold. Environment (MD: Manual Driving; AD: Automated Driving, SIM: Simulator, Synth: Synthetic); Image Modalities (R: RGB, I: NIR, D: Depth)

Dataset	**Year**	**Image Modalities**	**# Viewpoints**	**# Images (# Test)**	**# Keypoints**	**3D**	**# People**	**Environment**	**Annotation Method**	**Public**
2D general datasets										
MPII Human Pose [And14]	2014	R	-	40.5K (11.7K)	16	✗	-	Youtube	Manual	✓
MS Coco [Lin14] (Keypoints)	2016	R	-	165K (41K)	17	✗	-	Internet	Manual	✓
3D general datasets										
Shelf/Campus [Bel14]	2014	R	5	N/A	15	✓	6	Outdoor	Manual	✓
Human3.6m [Ion14]	2014	R/D	4	3.6M	24	✓	11	Lab	Marker Mocap	✓
CMU Panoptic [Joo16]	2016	R/D	520	1.5M	15	✓	N/A	Lab	Marker-less Mocap	✓
Itop [Haq16]	2016	D	2	100K	15	✓	20	Lab	automated + manual cleanup	✓
Automotive datasets										
Viva Challenge [Das15]	2015	R	6	11K (5.5K)	1	✗	N/A	MD	Manual	✓
Cronje[Cro17]	2017	R	N/A	N/A	4	✗	N/A	MD	manual	✗
Depth Pose [Mar17a]	**2017**	**I/D**	**1**	**20K (20K)**	**11**	**✓**	**9**	**Sim.**	**Manual**	**✗**
Chun[Chu19]	2019	I	2	18K (7.7K)	9	✗	100	MD	Manual	✗
Yuen[Yue19]	2019	R	1	10K (1.5K)	4	✗	N/A	MD	Manual	✗
MoLa R8.7k InCar[Bor20]	2020	I/D	1	8.7K (1.7K)	14	✓	5	MD	IMU & marker Mocap	✓
SVIRO[Dia20]	2020	R/I/D	1	25K (5K)	17	✗	32	Synth.	exported	✓
DriPe[Gue21]	2021	R	1	10K (2.6K)	17	✗	19	MD	Manual	✓
Drive&Act [Mar21]	**2021**	**R/I/D**	**6**	**6M (2D: 6K) (3D: 1.5K)**	**13**	**✓**	**4**	**AD**	**Automated (Manual)**	**✓**

Building on our experience collecting the first private dataset for 3D body pose estimation we extended the Drive&Act dataset with a public 3D body pose estimation benchmark by annotating a small subset of the images of the dataset [Mar21]. Compared to the other datasets, Drive&Act was focused on automated driving with a wide variety of secondary tasks causing occlusions and challenging poses. In contrast to Borges et al. [Bor20] the 3D driver body poses in our dataset were not instructed but were instead a product of the performed secondary activities. The Drive&Act dataset was recorded without an automated system for 3D driver body pose annotation. However, regarding Borges et al. [Bor20] and considering the high level of occlusion, caused by some secondary activities, automated annotation would have been challenging. Instead, we labeled 3D keypoint positions manually which was only possible with high accuracy because of the multi-view camera system of the Drive&Act dataset. We selected 1500 highly diverse scenes and annotated the upper body pose of the driver manually in four views of the dataset using triangulation to recover the 3D body keypoints. Overall, the dataset therefore consisted of 1500 scenes with 6000 manually annotated 2D driver body poses and the corresponding 1500 triangulated 3D driver body poses. In addition, Drive&Act provided six million 3D driver body poses for training, labeled with an automated process base on the triangulation of 2D driver body pose results determined with the OpenPose detector. Section 3.5.4 depicts the dataset and the annotation method in more detail. We evaluate the quality of the automatically annotated training set in Section 4.4.3.

2.2.2 Methods

There were many methods relying on driver body pose estimation to estimate distraction [Shi14], driver skill level [Tom12] or secondary activities [Xin18, Beh18b, Beh20, Beh18a] using either the Kinect SDK or other state of the art 2D body pose detectors.

However, research interest on driver body pose estimation itself only increased in recent years and public datasets were available for an even shorter time. There were other methods that detected just the hands [Mol15, Hoa17, Ran18] or the position and orientation of the head of the driver [Sch15, Sch17]. In the following we focus on methods that determine the location of all upper body keypoints of the driver.

Because of the depicted data scarcity methods that did not need training data combined with qualitative evaluations via example images were common initially. First, we summarize these approaches followed by methods that evaluated quantitatively and relied on state of the art machine learning methods.

To our knowledge one of the first methods for 3D driver body pose estimation on depth images was presented by Demirdjian et al. [Dem09]. They fitted an articulated mesh model to a low-resolution point cloud of the driver using an Articulated Iterative Closes

Point algorithm. Tran et al. [Tra10] used skin color segmentation to determine the head and hand positions in 3D using a stereo camera system. They approximated shoulder and elbow positions using inverse kinematics. Liu et al. [Liu13] proposed a method for 2D driver body pose estimation using Histogram of Oriented Gradients (HoG) features and a part template-based deformable model. Kondyli et al. [Kon15] used the Kinect Sensor to capture the driver. They noted that the Kinect SDK does not work reliably in real vehicles and therefore developed their own analytic approach. Yamada et al. [Yam16] also used the Kinect Sensor and presented a method based on body part labeling using Random Decision Forests to generate the 3D driver body pose. To simplify the collection of labeled training data, test participants wore a shirt with distinct colors for each body part. They then generated body part ground truth labels using chroma keying on the recorded color images. However, they still presented their results only qualitatively because their annotation method only resulted in labeled regions instead of 3D keypoints of the driver's body.

We contributed a similar Random Decision Forest-based method [Mar16, Mar17a]. However, while we also first collected a small real-world training set with body part labels, we later used synthetic data, like related methods from other domains. For testing, we collected data in a driving simulator with an automotive depth camera and annotated the data manually with 3D driver upper body keypoints. To our knowledge, this was the first published method for driver body pose estimation evaluated quantitatively as well as the first method using synthetic data to deal with the lack of annotated real data for driver body pose estimation (see Section 4.2).

In the following years deep learning-based methods became common which were trained on mostly private labeled datasets but with quantitative evaluation of their performance. Many approaches closely followed general 2D human body pose estimation methods regressing 2D keypoint positions [Oku18, Liu19b] or in most cases determining heatmaps of 2D keypoints [Gue21]. A popular framework was the multi-person body pose detector based on part affinity fields from the general research area [Cao17]. Methods based on this framework varied in the complexity of their base neural network to extract images features as well as in their input modalities, using color data [Chu19], depth images [Tor19, Bor20], near-infrared Images [Yue19] or color and motion data [Li18]. All of these methods determined just the 2D driver body pose even if they used depth images as input.

Estimating the 3D driver body pose was researched less often. Zhao et al. [Zha18] noted the unreliability of the Kinect SDK in the automotive context. They therefore created a post processing system to filter the results and to add missing joints. They used reliably detected joints to find close reference poses in a large body pose database to interpolate and correct the result of the Kinect SDK. Borges et al. [Bor20] used an approach for monocular 3D driver body pose estimatoin [Mar17b] to lift the 2D driver body pose to 3D with a deep fully connected network. Yao et al. [Yao20] proposed a two-stream approach using point clouds, created from depth images, in one stream and near-infrared images in a second stream to regress the 3D body pose of the driver.

Our updated method for 3D driver body pose estimation from depth images [Mar21] followed some of the presented trends by using a state of the art 2D body pose detector in conjunction with both near-infrared and depth images. However, we focused on fully decoupling 2D driver body pose estimation on near-infrared images and 3D driver body pose estimation via depth images. This had the advantage of utilizing large datasets for 2D body pose estimation, even from other domains, and could focus on lifting the results to 3D using less available and less varied annotated depth data from the car interior (see Section 4.3). In addition, we investigated different sensor setups for 3D driver body pose estimation via triangulation which was also a novel contribution (see Section 4.4.3).

2.3 General Activity Recognition

Activity recognition is a varied research field with many applications. Driver activity recognition is only a small part of this field of research. However, methods with an automotive context often originated from general activity recognition approaches. In the following we therefore present activity recognition datasets and methods in general as a reference for automotive approaches. We differentiate between video-based methods and 3D human body pose-based approaches, which were the basis of our work on driver activity recognition.

2.3.1 Datasets

There were large improvements in general activity recognition methods over the last decade. This can be attributed to the introduction of deep learning-based approaches as well as the availability of large public benchmarking datasets. Many activities have both spatial as well as temporal components. Most datasets and methods are therefore focused on video or other time series data. Datasets usually consist of short clips labeled with a single activity or longer video streams labeled in sections with different activities. These datasets are in general labeled manually. While the annotation process is the same for all datasets, there are differences regarding the collection of data depending on the input modality of the dataset for activity recognition.

Video-based activity recognition datasets, like datasets for 2D human body pose estimation, can be created from many sources like movies, TV shows or web videos. The datasets therefore depict activities with a large variety. HMDB [Kue11], UCF50 [Red13], UCF101 [Soo12] were the first large-scale video datasets for activity recognition with 50 and 101 classes respectively and many clips per class. The Kinetics dataset [Car17] is the current state of the art benchmark for activity recognition consisting of 400 classes at its introduction. In the following years the dataset was extended with even more activity classes and additional annotations as input for activity recogniton methods, like 2D body pose labels created with the OpenPose approach.

3D body pose-based activity recognition datasets face similar challenges as datasets for 3D human body pose estimation because the 3D body pose cannot easily be inferred from video data alone. However, the accuracy of the human body pose is less relevant as long as it depicts the performed activities well. This is why a popular choice to collect data was the Microsoft Kinect depth cameras in combination with the Kinect SDK for 3D human body pose estimation. Datasets were usually recorded in user studies with a limited number of people as well as instructed actions. Because of the simpler camera setup, compared to complex motion capture systems, the recording environments were more varied but still mostly limited to indoor spaces. However, the surroundings were less of a concern because the image data was not part of the input for these activity recognition systems.

The current benchmarks for this task are the NTU RGB+D 60 [Sha16b] and NTU RGB+D 120 [Liu19a] datasets. They include 60 and 120 activities, respectively. The datasets offer both a cross-view benchmark, where methods have to generalize to novel camera angles, as well as a cross-person benchmark, where methods have to generalize to unseen people. However, the datasets do not include additional annotations, for example, regarding the surrounding environment or objects involved in activities, thus methods solely focussed on the 3D human body pose as the input modality.

In contrast, the Drive&Act dataset, contributed by us for driver activity recognition, expanded the input domain for activity recognition to 3D object locations as well as a 3D model of the surrounding car interior.

2.3.2 Video-Based Methods

We used video-based activity recognition methods as comparison to our human body pose-based approaches both as benchmark in general as well as to highlight the advantages of human body pose-based methods with regards to changes in viewpoint and camera modality. In the following we therefore only introduce the research area in general and focus on the baseline methods used in our evaluation. For a detailed overview of the research area, we refer to Sun et al. [Sun21].

Deep learning-based video activity recognition methods mainly fall into three categories: Two-stream 2D convolutional neural networks, recurrent neural networks, and 3D CNN-based methods. 2D convolutional neural networks were often used together with a two-stream approach classifying color video data with one network and temporal data, in the form of optical flow, with a separate network using late fusion to combine the scores [Sim14]. Multi-stream approaches combining spatial and temporal data were also common for human body pose-based activity recognition. Methods based on recurrent networks often used 2D convolutional neural networks to extract features from each image followed by a recurrent network, based on long short term memory units (LSTMs), for activity classification [Don15].

3D convolutional neural networks can be applied to video classification by interpreting video sequences as spatio-temporal volumes. Tran et al. [Tra15] introduced the C3D network consisting of $3 \times 3 \times 3$ convolution layers and 3D pooling layers. The challenge with 3D convolutional networks was their increased number of parameters, introduced by the third dimension, which made them harder to train. Qiu et al. [Qiu17] therefore introduced Pseudo-3D Residual Nets (P3D) which mimic 3D convolutions with spatial 2D convolution ($1 \times 3 \times 3$) and temporal convolution ($3 \times 1 \times 1$). Carreira et al. [Car17] approached this problem differently, introducing Inflated 3D convolutional neural networks (I3D). They used a pre-trained inception network for image classification [Iof15] and inflated its weights to 3D. They achieved this by repeating the weights of 2D filters on the new third dimension and rescaling them to keep the output of the 3D filter the same as the 2D filter. With this approach it was possible to generate deep 3D convolutional neural networks with pretrained weights from image datasets.

We used C3D, P3D and I3D as baselines for our driver body pose based activity recognition methods (see Section 5.4.3).

2.3.3 3D Human Body Pose-Based Methods

Compared to video-based methods 3D human body pose-based approaches are more robust to noise introduced by the background, illumination changes or other objects. They can generalize well to other surroundings and viewpoints as demonstrated by [Liu19a] as well as by us (see Section 5.4.4). However, relying solely on the human body pose can also be a disadvantage if activities involve objects or interaction with the surrounding environment. The main evaluation datasets of the approaches presented here were the NTU RGB+D datasets [Sha16b, Liu19a] which were focused on 3D human body pose as the sole input modality. The research area therefore relied on human body pose data as its main input as well. Our approaches expanded these methods with additional data sources in an automotive context.

The main challenge for human body pose-based activity recognition methods was the efficient use of spatial as well as temporal relationships between body parts. This was usually approached in two stages: Creating suitable input features as well as parsing the spatio-temporal relationships with neural networks. Apart from the raw 3D keypoint trajectories, commonly extracted features were keypoint velocities, by subtracting keypoints of different time steps [Jan20], as well as body part orientations, by subtracting neighboring keypoints in the kinematic chain of the human body [Zha17b, Shi19a]. Often these features were combined in multi-stream neural networks with mid-level feature fusion [Jan20] or score-level late fusion [Shi19a].

Cross viewpoint evaluation was an important part of the NTU RGB+D datasets. The two most widely used methods to improve generalization to new views were either normalization [Sha16b, Shi20], by transforming the body pose data to a pose local coordinate system, or augmentation [Wan17, Zha19b], by perturbing the training data with random transformations. Instead of a fixed normalization scheme it was also possible to learn a normalizing transformation based on the input data as part of the neural network [Zha17a, Zha19b].

Even though the human body pose is already a high level, low dimensional representation, compared to video clips, there are still keypoints as well as parts of the time series that are more important than others. Attention methods enabled neural networks to focus on the important parts by re-weighting the features extracted by the primary neural network. There were attention modules for temporal [Son17a, Shi20], spatial [Son17a, Shi20] and feature channel [Shi20] importance estimation. There were also methods that refined a global state space using attention mechanisms [Liu17a, Mag19].

Approaches to parse spatio-temporal relationships differed widely depending on the base type of neural network. Recurrent networks were a popular choice at first because they were well suited to model the temporal dynamics of keypoint trajectories. However, modeling the spatial relationships between keypoints proved challenging. Graph convolution-based methods on the other hand were well suited for this task and mostly superseded methods based on recurrent networks. In parallel there were methods using advances in convolutional neural networks for image and video classification by casting the human body pose sequence in array like structures. In the following we present methods of all three areas.

Convolutional neural networks for image and video data require input in the form of arrays. They are well suited to extract features in local neighborhoods via convolutional filters. However, it is not obvious how to organize the sequence of human body poses in a grid with meaningful neighborhood because 3D human body movements have five dimensions (Keypoint type, Time and 3D coordinates). In addition, the limbs of the body form a tree structure which is hard to represent in a grid as well.

One way to achieve this was to project 3D keypoint coordinates onto the three main coordinate axes creating three scatter plots of keypoint positions. Temporal information as well as keypoint type could then be encoded using different ranges of the color spectrum [Hou16, Wan16b]. A more generalized approach used multiple permutation of two of the five dimensions for projection and encoded the remaining data as colors [Liu17b]. In all cases this resulted in multiple image-like matrices that were classified with separate convolutional networks followed by late fusion.

Stacking data without projection was the more common method leading to a compact matrix representation. This could be achieved by creating a vector of all keypoint data K from one time step and combining these vectors of different time steps T to a matrix of size KxT. While this approach worked well to represent the temporal neighborhood, the tree structured body model could not easily be mapped to a vector representing all spatial dependencies. This was tackled by grouping body parts based on limbs (e.g., arms, legs, torso) [Du15a, Ke17], training a permutation matrix to reorder body parts in a manner suitable for activity recognition [Li17], or by tree traversal keeping the local neighborhood consistent by replicating keypoints multiple times [Yan19, Cae19]. Traversal schemes were also a popular choice in conjunction with recurrent networks.

Recurrent neural networks are well suited to model time series data. Most approaches using recurrent networks for activity recognition based on 3D human body poses relied on Long Short Term Memory Units (LSTM) because they could model long term temporal relationships [Hoc97]. We also used LSTM units in our approach and introduce them in detail in Section 5.2.1. The baseline method to use recurrent networks for this task was stacking the human body pose data of each time step in a vector. The primary challenge for these methods was modeling spatial relationships between body parts. To this end there were two different approaches in research: Traversing the spatial structure of the input data and adapting the LSTM units themselves for activity recognition.

Du et al. [Du15b] concatenated keypoints in five parts corresponding to limbs and torso. They first extracted features from each part with separate LSTM units and fused these features pairwise in a hierarchy of additional LSTM Units. Liu et al. [Liu16] introduced a spatio-temporal LSTM. Their method used recurrency to travers the temporal as well as the spatial domain at the same time. They achieved this by assigning an LSTM Unit to each keypoint of the human body and by traversing spatially following the kinematic chain of the human body in depth first order. Spatial traversal followed the kinematic chain using depth first order. Wang et al. [Wan17] simplified this method using a two-stream approach where one branch extracted temporal features and the second branch used kinematic tree traversal to extract spatial features.

We based our first method for driver activity recognition on this framework. Our contribution was an extension with context features combining the body pose of the driver with the location of elements of the car interior, like the controls. It therefore allowed the method to reason about the context where an activity took place (see Section 5.1).

There were different proposed optimizations to the LSTM units themselves. Veeriah et al. [Vee15] proposed differential LSTMs taking the differential of the hidden state into account in the gates of the LSTM to make it more sensitive to temporal dynamics. Liu et al. [Liu16] introduced a trust-gate preventing cell updates if the keypoint position predicted from the hidden state differed from the input keypoint. Shahroudy et al. [Sha16b] introduced part aware LSTM cells by splitting the hidden state of the LSTM cell into five parts (e.g., limbs and torso). This reduced the number of trainable weights to update the cell state making the network more efficient.

Graph convolution-based neural networks (GCN) are well suited to represent the structure of the human body because the kinematic model naturally resembles a tree. Most methods were derived from the spatial graph convolution layer introduced by Kipf et al. [Kip17]. Their approach relied on vertex feature vectors and the graph neighborhood modeled as adjacency matrices. The graph convolution layer consisted of an aggregation step of neighboring node features, using the adiacency matrix and weighted averaging, followed by a trainable feature transformation. Yan et al. [Yan18] extended this approach to spatio-temporal graph convolutions (ST-GCN) for activity recognition on human body pose time series data by adding temporal convolutions, connecting each joint with its previous and next temporal neighbor.

We also relied on this framework for driver activity recognition and describe it in detail in Section 5.3.1. While the state of the art presented here mostly focused on optimizing ST-GCN using just human body pose data as input, we investigated how to expand the spatial graph to other data modalities like context features modelling the surrounding interior of the car as well as 3D object positions (see Section 5.3).

While ST-GCN only took the direct neighbors into account, relying on multiple stacked layers to propagate information through the graph, others accounted for neighbors further away [Li19a, Liu20, Hua20] be extending the adjacency matrix to second or third order graph neighbors. Others proposed changes to the temporal connections of the framework, extracting features at multiple temporal resolutions [Hua20] or creating additional temporal edges to nodes that are neighbors in the spatial graph [Liu20].

The adjacency matrix was the core of the graph convolution layer. It was often manually defined based on the kinematic model of the human body. In many cases this matrix was fixed for all layers of the network. This was restricting because it did not allow the machine learning model to optimize the graph structure. A popular choice was therefore to introduce additional trainable weight matrices of the same shape as the adjacency matrix. This matrix was either elementwise multiplied [Yan18], allowing the network to tune the importance of existing edges, or added [Shi19b, Hua20], allowing the network to create new edges. Others proposed to infer this matrix from the input data using separate neural network layers, allowing the graph to adapt to the current input [Li19a, Shi19b, Hua20, Ye20]. Instead of adding additional matrices it was also possible to use the manually defined graph only to initialize the adjacency matrix and to adapt it as part of the training process [Shi19a]. Training the adjacency matrix from scratch with randomized initialization was challenging and often unstable. The manually defined adjacency matrix stabilized the training process [Shi19a] and was therefore often included in some form. Most of the presented approaches were used to optimize a single global adjacency matrix for all layers but it was also possible to apply these techniques to each layer separately [Shi20].

A drawback of the basic graph convolution layer was its missing definition of structure in the neighborhood. A 3×3 convolution layer for images for example has nine filter weights, one for each of the pixels. In graph convolution layers there was only one trainable weight for all neighbors because the order of neighbors in the graph is undefined. Yan et al. [Yan18] therefore defined a neighborhood based on the distance to the center of the human body. They could then split the edges of the graph into three parts with separate trainable weights. However, this approach increased the needed computational resources proportional to the size of the neighborhood. Cheng et al. [Che20b] therefore proposed to apply different trainable adjacency matrices to partitions of the node feature vector keeping the needed resources constant.

2.4 Driver Activity Recognition

We conducted a review of the state of the art for driver activity recognition, like our literature review on driver body pose estimation. We discovered relevant publications by searching for keywords related to driver activity recognition. In addition, we reviewed all papers that cited public datasets for the task. The resulting list was filtered for contributions that either published datasets for driver activity recognition or propose a method for the task. Overall, we discovered 91 relevant papers from 2012 until the end of 2021 (see Figure 2.3). To the best of our knowledge this represents the main body of work on driver activity recognition in the last decade. We could identify some general trends. Research interest increased strongly since 2018. However, compared to general activity recognition large-scale public datasets only became available in the last years, in part thanks to our contribution of the Drive&Act dataset [Mar19]. The lack of large-scale data also delayed the adoption of deep learning-based methods. While general activity recognition methods

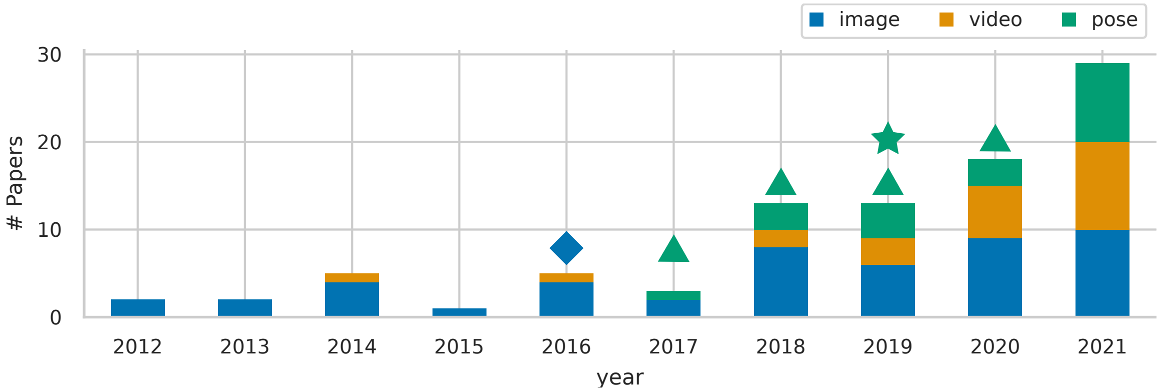


Figure 2.3: Number of papers per year for driver activity recognition. ▲ indicates own publications. ★ indicates own public datasets. ◆ indicates the first public dataset.

mostly relied on time series data, as shown in the last sections, image-based methods were still common for driver activity recognition, which was likely also an effect of the limited data availability. Driver body pose-based activity recognition, the focus of our work, became popular with the introduction of general human body pose-based methods that worked well across different domains making training on automotive data less necessary.

In the following section, we present our findings in detail starting with dataset, followed by methods detecting interaction with the interior, like hands on wheel detection, followed by activity recognition methods in three sections based on images, videos or the body pose of the driver.

2.4.1 Datasets

Compared to driver body pose estimation public datasets were more common for driver activity recognition. Nevertheless, until recently these datasets were limited in size and complexity. The established benchmark datasets were image-based and mostly depicted clear-cut samples for each activity [Kag16, Abo18]. At the time of writing there are no established benchmark datasets for video or driver body pose-based activity recognition with an automotive focus. Private datasets, that were only used in a few publications, are common and the recently released large-scale datasets are not established as common benchmarks yet. Table 2.2 shows an overview of some private and all public datasets, discovered in our literature search, in comparison to related datasets from other domains.

To our knowledge Zhao et al. [Zha12a] presented the first dataset for driver activity recognition that was used in multiple publications over the years. The first iteration of the dataset consisted of images extracted from the recorded videos with four activity classes. The dataset was later extended to six classes [Yan16c] as well as to video data [Yan16a]. However, according to Abouelnaga et al. [Abo18] the data was not publicly available.

The first public and the most used datasets for driver activity recognition were the STATE-FARM dataset [Kag16] and the AUC dataset [Abo18, Era19]. The data was annotated with 10 different activities. However, both datasets consisted of images extracted from the recorded video data and showed mostly iconic scenes that represented the activity classes well with few ambiguities. The datasets are still widely used despite their small size.

Methods that relied on time-series data could not use these datasets. There were various publications based on small datasets [Yan14, Bil19, Nav21]. The first large public video dataset related to driver activity recognition was the Brain4Cars dataset with $2M$ images [Jai15b]. However, the purpose of the dataset was driver maneuver prediction.

For driver activity recognition, to our knowledge, we contributed the first method that was evaluated on a dataset of more than $1M$ images (see Section 3.4) [Mar18b]. It was also the first to rely on depth images for 3D driver body pose data. However, we were not able to publish the data for privacy reasons. In the following year we published the Drive&Act dataset [Mar19] which surpassed previous datasets in every area by a large margin, reaching the size of activity datasets from other domains. In addition, it was the first dataset that focused on activities for automated cars (e.g., watching movies or working on a laptop). Overall, it contained 83 activity classes as well as 3D driver body pose annotations for driver activity recognition (see Section 3.5). We extended the dataset in the following year with object bounding boxes and object 3D positions [Mar20a]. Reiss et al. [Rei20b] extended the dataset for Zero-Shot driver activity recognition and Roitberg et al. [Roi20b] extended it for open-set driver activity recognition.

With the rising interest in driver activity recognition the number of more complex and public datasets also increased rapidly. In the same year we published the Drive&Act dataset, Jegham et al. [Jeg19, Jeg20] published the MDAD dataset. It focused on manual driving providing hand and face bounding boxes in addition to 16 activity classes. Ortega et al. [Ort20] proposed the DMD dataset. It is the largest dataset for driver activity recognition while driving manually, to date. They provided various annotations including labels for the visual focus of attention of the driver. However, only a part of the dataset was made public yet [Cañ21]. They split data acquisition into multiple parts, recording in a simulator, outside in a parked vehicle as well as while driving, depending on the risk of the performed activities. Katrolia et al. [Kat21] proposed a dataset consisting of synthetic rendered data as well as real data recorded in a simulator. They provided 3D bounding boxes and instance masks for objects and the driver. Lin et al. [Lin21] recorded a dataset using the Kinect and the Kinect SDK for 3D driver body pose estimation. They recorded in a low fidelity driving simulator. Apart from Drive&Act this is the only other dataset with 3D driver body pose annotations. Tan et al. [Tan21] published a large dataset for bus driver activity recognition.

Table 2.2: Summary of datasets for activity recognition. Video and 3D general datasets are provided as reference. Automotive datasets are a complete list of existing public datasets as well as important private datasets. Own contributions marked bold. Environment (MD: Manual Driving; AD: Automated Driving); Image Modalities (R: RGB, I: NIR, D: Depth)

.

Dataset	Year	Image Modalities	# Viewpoints	# Images	# Actions	Body Pose	# Objects	# People	Environment	Video	Public
Video Datasets											
HMDB-51 [Kue11]	2011	R	-	0.5M	51	✗	✗	-	Movies, Youtube	✓	✓
Kinetics 400 [Car17]	2017	R	-	76M	400	✗	BB	-	Youtube	✓	✓
3D Human Body Pose Datasets											
NTU-RGB+D 60 [Sha16b]	2016	R/I/D	3	4M	60	3D	✗	40	Indoor	✓	✓
NTU-RGB+D 120 [Liu19a]	2019	R/I/D	3	8M	120	3D	✗	106	Indoor	✓	✓
Automotive Datasets											
SEU[Zha12a]	2012	R	1	N/A	4(6)	✗	✗	20	MD	✗	✗
Yan et al. [Yan14]	2014	R	1	N/A	5	✗	✗	20	MD	✓	✗
Brain4Cars [Jai15b]	2015	R	2	2M	5	✗	✗	10	MD	✓	✓
D.P.-Night [Yan16b]	2016	R/I	1	29K	4	✗	✗	20	MD	✗	✗
D.P.-Real [Yan16b]	2016	R	1	18K	4	✗	✗	5	MD	✗	✗
Statefarm [Kag16]	2016	R	1	22.5K	10	✗	✗	NA	MD	✗	✓
AUC-D.D. [Abo18]	2018	R	1	14.4K	10	✗	✗	44	MD	✗	✓
Incarin Activity [Mar18b]	**2018**	**I/D**	**1**	**1.3M**	**7**	**3D**	✗	**26**	**MD test track**	✓	✗
Drive&Act [Mar19]	**2019**	**R/I/D**	**6**	**>9.6M**	**83**	**3D**	**BB, 3D**	**15**	**AD Simulator**	✓	✓
EBDD[Bil19]	2019	R	1	70K	5	✗	✗	13	MD	✓	✓
MDAD[Jeg19]	2019	R/D	2	7h	16	✗	BB	50	MD	✓	✓
DMD[Ort20]	2020	R/I/D	3	41h	13	2D	BB	37	MD, Sim	✓	✓
TICaM[Kat21]	2021	R/I/D	1	126K	19	2D	BB, 3D	13	MD, Sim, synthetic	✓	✓
Driver-Skeleton [Lin21]	2021	R/D	1	100K	10	3D	✗	30	MD, Sim	✓	✓
PCL-BDB [Tan21]	2021	R/D/I	1	N/A	11(40)	2D	✗	55	MD, Bus	✓	✓
UET[Nav21]	2021	R	1	23.7K	11	2D	✗	10	MD	✓	✓

2.4.2 Methods to Detect Interaction With the Interior

We define driver interior interaction as a sub-task of driver activity recognition with the primary purpose of detecting if the driver is interacting with parts of the interior, like the steering wheel, the gear stick or the infotainment interface. These activities are also part of some general driver activity recognition datasets, but this section will only highlight methods with a focus on this sub-task. Compared to other activities the surroundings like the position of the steering wheel or the gear stick have a direct relationship with the corresponding activity. This highlights the need to incorporate features of the surroundings for their detection which was investigate in detail by our driver activity recognition approaches.

Some methods modeled the location of different elements explicitly. Ohn-Bar et al. [Ohn13] defined bounding boxes on the image of a camera mounted at the ceiling and trained support vector machines (SVMs) for each bounding box to detect the presence of hands. Hoang Ngan Le et al. [Hoa16] used a deep learning-based object detector to detect the hands and the steering wheel and used the intersection of these bounding boxes to determine steering wheel interaction.

Our own method for the task relied on the 3D body pose of the driver [Mar17a]. We manually defined the location of interior elements in 3D using geometric primitives, like cubes and cylinders. To estimate interaction with these areas we determined the distance of the hands to the surface of these volumes (see Section 5.1). Compared to previous related methods our approach could be configured for arbitrary interior elements and different cars without any retraining.

Perrett et al. [Per17] determined if the driver or the co-driver interacted with the center console by extracting hand silhouettes in the area using background subtraction. They also determined how the passengers interacted with the area by analyzing the silhouettes for different hand poses. Borghi et al. [Bor18] proposed a system for detailed steering wheel interaction detection. They used a stereo camera behind the steering wheel to determine both the hand positions on the wheel and if the driver is actually grasping the wheel.

Instead of using the location of interior elements explicitly others relied on the image data alone using various machine learning methods. Xu et al. [Xu14] used random decision forest to determine interaction with five zones. Siddharth et al. [Sid16] used an object detector to determine the position of the hand and classified the extracted hand patch using HOG features and SVMs to detect the grasping of the steering wheel. Gu et al. [Gu22] similarly used an object detector but instead of classifying the extracted hand patch separately they directly determined three different bounding box classes, grasping the steering wheel, idling and performing other tasks.

2.4.3 Image-Based Methods

Image-based methods for driver activity recognition rely on just a single image as input. They do not account for the motion of different activities. This was still a popular approach for driver activity recognition while the related work for general activity recognition was mostly focused on video data and other time series data. The reason was likely the availability of public datasets which were largely image-based until publication of our Drive&Act dataset, among others. The common benchmarking datasets for these methods are SEU, STATEFARM and AUC (see Table 2.2).

Classic machine learning approaches using hand-crafted features and classifiers like SVMs could already achieve good performance on these dataset. Popular features were: Skin color segmentation [Zha12a, Zha12b, Zha13, Gup15, Yan16c, Era19, Xin19], Histograms of Oriented Gradients (HOG) [Zha13, Hss17, JEG18, Are19, Qin21] and keypoint detectors like Sift [Hss17, Ber21] or Surf [JEG18]. These features were then classified with SVMs [Hss17], Multi-Layer Perceptrons [Zha12b, Zha13, Gup15], Random Decision Forests [Zha12a, Maj18] or even CNNs [Yan16c, Xin19, Qin21].

CNN-based approaches for image classification were also often applied for image-based driver activity recognition using Alexnet [Hss17, Abo18], VGG [Bah18, Mas18, Kap20], ResNet [Hss17], Inception [Kap20, Maf20, Wan21b] or Mobilenet [Kap20, Cañ21]. Some also designed their own neural networks [Alo20, Jam20] using the principles of these approaches like skip connections [Ngu21], multiple filter resolutions [Hu19], separable convolutions [Bah20, Ngu21] or attention mechanisms [Hu20].

There were also methods combining handcrafted features with CNNs for classification. For example, by extracting image features with HOG as well as Alexnet and classifying the concatenated features with fully connected layers [Are19] or by extracting HOG features from the whole image and classifying them with CNNs [Qin21].

Another trend was pre-processing of the image before classification to suppress the background or to extract areas of interest from the image using skin color segmentation, grab cut [Lee19] or body part labeling via CNNs [Ezz21] to create a segmentation of the driver. Others used object detectors to extract the driver area [Wan21b, Zha21], head or hands [Abo18] or activity area [Lu19] from the image.

While there were many approaches for image-based driver activity recognition it is hard to evaluate the progress in this area. The reason for this lies in the datasets and the accuracy already achieved by the baseline methods proposed by the dataset authors which left only a narrow margin for future improvement. Zhao et al. [Zha12a] proposed the SEU dataset. Their method involved skin color segmentation with a Gaussian Mixture Model followed by Contourlet transformation and classification with Random Decision Forests.

They already achieved 90.5 % accuracy. Abouelnaga et al. [Abo18] published the AUC dataset. Alexnet trained on the dataset already achieved an accuracy of 93.65 %. Their proposed ensemble of multiple models reached 95.95 % accuracy. Hssayeni et al. [Hss17] tested different handcrafted features and neural networks on the STATEFARM dataset. They could show that HOG features and SIFT features combined with SVMs do not perform well on the dataset. Alexnet on the other hand already achieved 72.6 % accuracy and ResNet-152 reached 85 %. In comparison our best baseline method on Drive&Act achieved just 63 % accuracy.

2.4.4 Video-Based Methods

Video-based driver activity recognition methods need large amounts of video data for training. Some methods relied on the previously discussed image-based datasets by organizing them into sequences according to the videos they were extracted from [Val18, Mos19, Che20a, Nel21]. However, most either used private datasets or the recently published large-scale driver activity datasets like Drive&Act.

Methods relying on hand-crafted features often combined spatial image features as depicted in the last section with temporal features, like motion history images [Yan14], motion intensity images [Yan16a] or space time interest points (STIP) [Jeg19] and used Random Decision Forests [Yan14] or SVMs [Yan16a, Jeg19] for classification.

Deep learning-based driver activity recognition methods mostly followed the approaches for general driver activity recognition using two-stream 2D-CNNs, recurrent networks or 3D-CNNs.

Driver activity recognition systems using recurrent networks usually combined image features, extracted via 2D-CNNs, with LSTM units [Jeg20, Cañ21]. Some extended this framework using foreground masks created by segmenting a corresponding depth image [Jeg21] or by introducing attention mechanisms for image features as well as temporal dynamics [Wha21].

Two-stream methods extracted features with 2D-CNNs in two streams using video data and optical flow as input [Hu18, Che20a]. Yang et al. [Yan21] expanded this approach by first using an object detector to crop the video sequence to the relevant driver area. Instead of two separate streams Kose et al. [Kos19] stacked optical flow and image data and inferred activities with a single model.

3D CNN-based approaches were by far the most common [Ort20, Nel21, Cañ21]. The I3D architecture was often used as the basis of these methods [Val18, Mos19]. We also used I3D as baseline model for Drive&Act [Mar19]. This network was often pre-trained on the Kinetics dataset and only fine tuned on the limited automotive data. Some also limited fine tuning to the last layers of the model because of the limited size of their data [Mos19]. Liu et al. [Liu21] proposed a method to improve global feature extraction using multi-task learning. They generated positive and negative samples from the input sequence in addition to the original data. They then trained their model to classify activities as well as to maximize the distance of the feature space between positive and negative video samples. Ren et al. [Ren21] focused on the activities of the Drive&Act dataset involving object by fusing the results of an activity detector and an object detector using majority voting.

Apart from supervised driver activity recognition the larger driver activity datasets published in recent years allowed to explore related task important for future driver monitoring systems like driver anomaly detection [Kop21], uncertainty analysis and calibration of driver activity recognition methods [Roi20a, Roi21], zero shot driver activity recognition [Rei20b], open set driver activity estimation [Roi20b] or domain adaptation [Rei20a]. The Drive&Act dataset published by us proved to be a popular starting point for these research directions going beyond supervised closed set classification of driver activities.

2.4.5 Driver Body Pose-Based Methods

This section outlines all driver activity recognition approaches that include driver body pose estimation as input feature. However, in many cases methods also used other data sources as input for their activity recognition system including image data, hand pose, head pose and objects. None of the methods apart from our own investigated the use of an interior model, describing the position of controls and other elements, for driver activity recognition. The most common body pose detector by a large margin was OpenPose (see Figure 2.1), which we also used. Usually, the detector was not trained on automotive data but instead on the MS-COCO dataset. While this worked well, based on visual inspection, we contributed an objective evaluation of the performance of OpenPose on the Drive&Act dataset (see Section 4.4.1).

In the following we present these methods in three sections according to our previous structure based on the input format of the method: using just one image or time step, using video or time series data, or using 3D driver body poses.

Image-based methods that rely on the driver body pose were often extensions of the previously presented methods using just image classification. They were also tested on the same datasets (STATEFARM and AUC).

Behera et al. [Beh18a] used a CNN with DenseNet architecture to classify driver activities from the image. They integrated the driver body pose via mid-level fusion of the heatmaps of the body pose detector and the features extracted by the CNN. Behera et al. [Beh20] extracted the 2D driver body pose as well as object positions in pixel coordinates. They determined the angle and pixel distance of keypoint pairs as well as keypoint object pairs to create feature histograms. They combined these histograms with image features, extracted via CNN, for the final activity classification. Çetinkaya et al. [Çet21] used Random Decision Forests to classify the driver body pose as well as a CNN to classify the image. They combined the scores with weighted score fusion to determine the final result. Koay et al. [Koa21] determined the body pose and the hand skeletons of the driver. They rendered these features as a stick figure onto an image. To determine activities they used a two-stream approach extracting features from the original image and the rendered image with mid-level feature fusion. Wu et al. [Wu21] determined image features with a CNN, body pose features from the keypoints of the drivers body with a multi-layer perceptron and hand features from a cropped image around the wrist keypoints with a CNN. They combined these features with mid-level feature fusion for the final classification.

Video-based methods extract the 2D body pose from video data. Most methods followed the trends depicted for general human body pose-based activity recognition either using recurrent networks based on LSTM units or using Graph Convolutions-based on the ST-GCN framework [Yan18].

Behera et al. [Beh18b] extracted features per image with a CNN as well as keypoint pair histogram and keypoint object histogram features. They classified each feature with a LSTM layer and combined the output with a second LSTM layer. Jiao et al. [Jia21] first determined keyframes of the input video sequence by clustering the body pose with K-Means Clustering. They then classified the keyframe sequence with an LSTM-based network.

Li et al. [Li19b] used ST-GCN on the 2D driver body pose to determine activities. They introduced a genetically weighted voting system to combine classification scores of different time steps based on their importance. Pan et al. [Pan21] used spatial graph convolution for each time step to extract features and applied an LSTM to classify the resulting feature sequence. Tan et al. [Tan21] tightly combined image and body pose features. They used a two-stream approach where the first stream extracted features from the image with a CNN and the second stream used ST-GCN to extract features from the driver body pose sequence. They enriched the mid-level node features of the ST-GCN network with mid-level features from the image stream. They then used the final output of the the body pose stream as attention module for the classification of the final image features. Wang et al. [Wan21c] divided the driver body pose sequence into multiple overlapping segments. They used ST-GCN to extract features from each segment and combined the results with an LSTM layer for final classification.

3D driver body pose based methods need either multiple cameras or depth cameras to determine 3D keypoint positions of the driver's body. Like our approaches most methods relied on depth cameras to determine the 3D driver body pose, however, to our knowledge, we are the only ones to also use a multi-view system evaluating both cross-view and cross-modal performance of our driver activity recognition approach.

To our knowledge we published one of the first methods for driver activity recognition based on 3D driver body pose estimation [Mar18b, Mar19]. Our method relied on three recurrent streams parsing the temporal as well as spatial dynamics of the drivers movements and in addition the distance of body parts to elements of the interior, like the steering wheel. We combined these streams with weighted score-level fusion (see Section 5.2).

Xing et al. [Xin18] used Random Decision Forests to determine the importance of each element of the 3D driver body pose as well as head pose. They used the most important elements for classification with a multi-layer perceptron. Weyers et al. [Wey19] extracted regions in a near-infrared image around the hand keypoints of the skeleton. They generated image features from these patches with a CNN and classified the body pose concatenated with these features using an LSTM network. This gave their method the ability to consider objects and hand poses that were by the body pose of the driver.

We took another approach with a similar result and introduced 3D object positions to our driver activity recognition system [Mar20a]. This way our activity recognition method was still fully independent from image data. We cast the 3D driver body pose, 3D interior elements as well as 3D object positions into a sparse spatio-temporal interaction graph and used ST-GCN to infer activities based on this representation (see Section 5.3).

Lin et al. [Lin21] used spatio-temporal graph convolutions in two streams using keypoint position and keypoint velocity as input features. They also introduced a temporal attention module for driver activity recognition.

2.5 Summary

This chapter outlined the progress of general as well as automotive human body pose estimation and activity recognition methods over the last decade. For each section we highlighted the research gaps we could identify and our contributions regarding datasets as well as methods.

There were large general datasets for both areas at the start of the thesis. However, for automotive applications public data was not available. We first contributed methods, tested on private datasets larger than other automotive approaches [Mar17a, Mar18b], followed by our publication of the Drive&Act dataset as one of the first large-scale driver activity benchmarks [Mar19]. In the following years we extended this dataset with object bounding box and 3D annotations [Mar20a] as well as benchmarks for 2D and 3D driver body pose estimation [Mar21]. In addition, the dataset was extended by Reiss et al. [Rei20b] for zero shot driver activity recognition and by Roitberg et al. [Roi20b] for open set activity recognition. While there are now other large driver activity datasets, Drive&Act is still the only dataset focused on automated driving and, with all extensions, provides the most varied set of benchmarks. There are still no large-scale datasets for driver body pose estimation but there are small datasets similar in size to the benchmark we provided for Drive&Act. However, our benchmark is extracted from the complex activity classes of the Drive&Act dataset with varied driver body poses and challenging occlusions.

While there is a longer history for driver body pose estimation the first methods were limited by the lack of annotated data and were therefore often hand-crafted and evaluated on example images. To our knowledge, we provided the first method for 3D driver body pose estimation evaluated objectively on annotated data as well as trained on synthetic data [Mar16, Mar17a]. In the following, deep learning-based methods for general 2D human body pose estimation also became popular for driver body pose estimation. However, research on 3D driver body pose estimation remained less popular, likely because of the lack of data and the challenging data annotation process. We tackled this problem with our second approach by splitting the task into 2D body pose estimation on color or near-infrared images and 3D body pose lifting via depth images. This allowed us to use large-scale general datasets for 2D body pose estimation in addition to the limited automotive data sources [Mar21]. In addition, we contributed an evaluation of multi-view 3D driver body pose estimation methods.

General 3D human body pose-based activity recognition systems mostly focused on just the body pose as input. In the automotive context body pose-based methods became popular much later but also incorporated objects in some cases. However, most methods focused on 2D driver body pose data as well as object bounding boxes and in some cases image features. Our contributions focus on 3D data without additional image features. We extended general methods based solely on the human body pose with extra features regarding the surrounding interior [Mar18b, Mar19] as well as 3D object positions [Mar20a]. In contrast to other automotive methods, we also showed that our approach can generalize across different views and sensor modalities.

3 Datasets for Driver Monitoring

To achieve the goals of this thesis with regards to 3D driver pose estimation and activity recognition, suitable datasets for the proposed methods were necessary. As outlined in our review of related work there were no suitable datasets at the start of the thesis. This made it necessary to collect as well as publish datasets as part of our work.

This chapter first systematically depicts the challenges and solutions for monitoring the two front seats inside the car. This includes a review of suitable sensor systems to monitor the passengers in 3D in all lighting conditions, followed by an evaluation of different camera mounting points regarding field of view and susceptibility to occlusions.

On this basis we present the datasets that were collected in the context of this thesis. Each of the presented datasets was built on the experience gathered while recording and using the previous datasets. The first two datasets focused on specific parts of the developed algorithms starting with 3D driver body pose estimation (see Section 3.3) and followed by driver activity recognition (see Section 3.4). This led to our final dataset being published under the name Drive&Act (see Section 3.5). It combined and extended all challenging aspects of our previous datasets. It was used for the final evaluation of all methods presented in the following chapters.

3.1 Camera Selection for 3D Cabin Monitoring

Selecting a camara system for driver monitoring is a complex problem because the interior of the car poses some unique challenges. It is a confined space with limited mounting positions for cameras, making a large field of view necessary in order to capture the driver or even both front passengers (see Section 3.2). This is challenging for some sensor technologies. In addition, the lighting conditions within the car can range from almost complete darkness at night to bright sunlight. Developing a camera system that is well tested in all conditions was out of the scope of this thesis. However, we kept these constraints in mind and selected sensors that would work at night using near-infrared illumination as well as in bright daylight.

The last requirement for the camera system was the ability to create 3D data either with the camera itself or in following steps using triangulation-based 3D reconstruction methods. The camera systems that fit these requirements were depth cameras with a single sensor using structured light or time of flight (TOF) approaches as well as stereo or multi-camera setups. In the following we first depict why interior monitoring systems use near-infrared light and its challenges before discussing the capabilities of different sensor technologies for depth sensing, including the sensors chosen for collecting our datasets.

Near-infrared Light (NIR) Illumination for driver monitoring

Optical interior monitoring systems need a light source to work at night. However, its light should not disturb the passengers, for example by interfering with the night vision of the driver. Illumination in the visible spectrum (380 to 800 nm) is therefore not an option. Although there are sensors for almost any wavelength of light, small and affordable cameras are mostly optimized to work in the visible light spectrum. Their efficiency to capture light usually drops sharply for other spectra (see Figure 3.1). In addition, shorter wavelengths in the UV spectrum (<380 nm) can be harmful to the human skin and eyes. The best illumination option is therefore near-infrared light (NIR) with a wavelength of 800 to 1000 nm. It is therefore used for surveillance cameras, eye tracking applications and other driver monitoring systems as well as depth cameras based on structured light or time of flight methods. The most common wavelengths used in these products is 840 to 860 nm because of the higher sensor sensitivity compared to longer wavelengths. However, this wavelength is still visible to the human eye as a red glow. For this reason, commercial automotive systems rely on illumination with a wavelength of 940 nm accepting an additional penalty to the efficiency of the sensor.

While near-infrared cameras need illumination to work at night, many depth cameras also rely on their light source to measure distance, which can be a challenge in bright sunlight because the emitted intensity is much less than the intensity of the sun. Bandpass filters are one counter measure that can be applied to any camera system to block most of the sun's electromagnetic radiation outside the spectrum emitted by the active illumination system. In addition, cameras with high dynamic range are preferred for automotive applications. Modulation of the intensity of the light source can also help to suppress light from other sources. This method is often used by time of flight cameras that also rely on modulation to measure depth.

Near-infrared illumination can be harmful to the human eye because natural reflexes do not work for spectra that are invisible to the eye. The primary health concern with near-infrared light is heating of the cornea and retina. To limit the exposure to safe levels and still provide good illumination, most systems use high power pulsed light sources to provide illumination only when images are taken. Because of their potential risk the illumination systems need to be tested for eye safety.

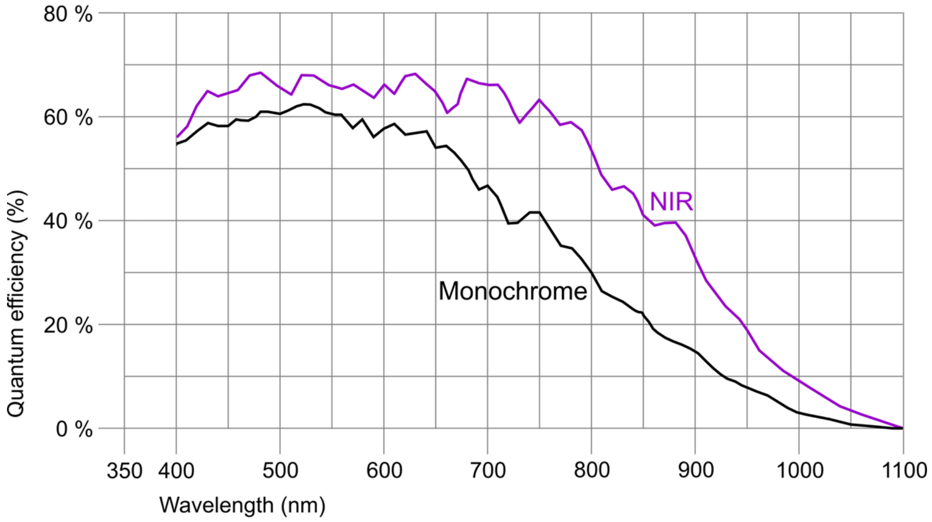


Figure 3.1: Quantum efficiency curve of the near-infrared optimized sensor (UI-3241LE-NIR-GL[2]) used to record Drive&Act compared to the monochrome version for visible light.

Structured Light Depth Cameras

The basic principle of structured light systems is the reconstruction of a 3D scene or object by projecting a known light pattern and measuring the changes resulting from the surfaces of the scene with one or more calibrated cameras. The principle is used in many different applications ranging from 3D scanning of objects, measuring deformation of surfaces in industrial applications to human computer interaction.

Pseudorandom points are a common pattern for high quality real-time capable structured light systems. They reconstruct the scene by matching blocks of random dots to a reference pattern and measuring shift as well as deformation of the pattern. We disregard this technology for our data collection efforts because it suffers from various drawbacks. The near-infrared sensors of these cameras are optimized to capture the projected pattern, often leaving the surrounding area underexposed. The bright dots in the image make it unsuitable for many image-based deep learning methods. In addition, the pattern is often no longer visible in sunlight, making depth reconstruction impossible. To arrive at this conclusion, we tested both the Microsoft Kinect for Windows [DiF15] as well as the Orbbec Astra[3] in darkness and in sunlight.

[2] https://en.ids-imaging.com/IDS/datasheet_pdf.php?sku=AB00432, accessed: June 14, 2022

[3] https://shop.orbbec3d.com/Astra, accessed: June 14, 2022

Name	**Melexis MLX75023**
Type	Time of Flight
Illumination	NIR Laser (940 nm)
Resolution	320 px × 240 px, 30 fps
Min. Range	0.1 m
FoV	H: 86.5° V: 69.3°
Size (W/H/D)	135 mm × 105 mm × 30 mm

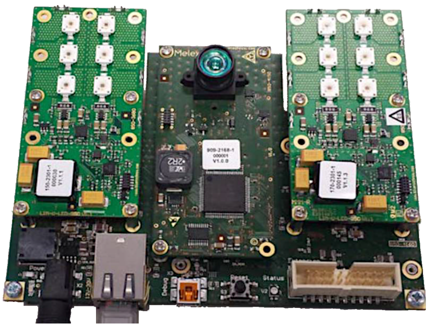


Figure 3.2: Specifications of the Melexis ToF Evaluation Kit.

Time of Flight (TOF) Depth Cameras

Cameras relying on the time of flight principle are another option to generate depth images with a single sensor. The approach relies on measuring the time light needs from a light source controlled by the camera to the sensor via reflection in the scene. This technique requires specialized circuitry both for the modulation of the illumination system as well as for the TOF camera sensor [Kol10]. The method has the advantage of producing both a depth image and a near-infrared image from a single sensor resulting in perfect alignment of the image data. In addition, because of the employed modulation techniques, these sensors can block other light sources, such as the sun, to a large degree. The near-infrared image is therefore robust to brightness changes as well as uneven lighting caused by shadows. However, sunlight still affects these sensors. With increasing brightness from other light sources, the noise level of the depth image increases up to a point where depth estimation fails. The primary drawbacks of these cameras are their often limited resolution, limited field of view as well as their minimum range requirement below which no depth data can be provided (e.g., 0.5 m for the Kinect 2 for Windows). This can restrict where the camera can be placed in the interior of the car.

Schwarz [Sch18] tested different consumer TOF cameras for their suitability for driver head pose detection. They determined that the Kinect 2 for Windows (see Figure 3.3) works well for automotive applications even in sunlight. We followed their findings and used this camera to record some of our datasets. In addition, we used the Melexis MLX75023 sensor (see Figure 3.2) which is an automotive sensor also used in production cars[4]. It is robust to sunlight according to the manufacturer. Although it has a lower resolution than the Kinect it has a larger field of view and shorter minimal distance requirement which allowed us to select a better position for the sensor in the interior.

[4] https://www.melexis.com/en/tech-talks/enabling-potential-automotive-3d-tof-imaging, accessed: June 14, 2022

Name	Kinect 2 for Windows
Type	Time of Flight
Illumination	NIR Laser (850 nm)
NIR Filter	Bandpass 850 nm
Resolution	512 px × 424 px, 30 fps
Min. Range	0.5 m
FoV	H: 70° V: 60°
Size (W/H/D)	250 mm × 65 mm × 85 mm

Figure 3.3: Specifications of the Kinect 2 for Windows ToF camera.

Stereo and Multi-View Systems

Apart from special sensors for depth perception, multiple conventional cameras can also be used together with triangulation methods to generate depth data. However, active illumination is still necessary for these systems to work at night. The advantage of conventional cameras is their great flexibility, low cost, and great potential for miniaturization.

Although multi-camera systems offer great flexibility, generating depth images or point clouds is computationally expensive. Compared to the previously discussed camera systems for depth perception, stereo cameras are often less accurate for indoor use. In addition, they have difficulties reconstructing texture-less surfaces because it is difficult to find point correspondences in these areas. Instead of computing dense depth images or point clouds based on multi-view data we therefore applied detectors on each image and relied on triangulation to reconstruct 3D data just based on detection results. With this approach we were able to determine the 3D driver body pose by triangulating the results of a 2D body pose detector (see Section 4.3.2). The great advantage of multi-view systems is their increased robustness to occlusions via redundant views which we used to generate ground truth data for 3D driver body pose estimation (see Section 3.5.4).

According to our requirements we designed a small monocular camera system with interchangeable lenses, a bandpass filter to block external light and a near-infrared LED (Light Emitting Diode) ring light with a wavelength of 850nm. All components were integrated using a custom 3D printed housing (see Figure 3.4). We used five of these camera systems in a calibrated multi-view setup to record the Drive&Act dataset. Both a single camera system as well as the multi-view setup were tested externally for eye safety according to DIN EN 62471:2009. We chose a continuous illumination system instead of a pulsed one to simplify the eye safety requirements with regards to circuitry for eye safety in failure cases. This was only possible because of the chosen near-infrared light optimized sensor (see Figure 3.1). The multi-view system included four cabin cameras with a wide field of view and a driver head monitoring camera with a narrow field of view. We discuss the selection of the camera positions and the field of view of the lenses in the next section.

Name	**IDS UI-3241LE-NIR-GL**
Type	NIR Optimized Grayscale
Illumination	NIR Ring Light (850 nm)
NIR Filter	Bandpass 850 nm
Resolution	1280 px × 1024 px, 60 fps
FoV Cabin	H: 138° V: 104°
FoV Head	H: 47° V: 38°
Size (W/H/D)	55 mm × 55 mm × 40 mm

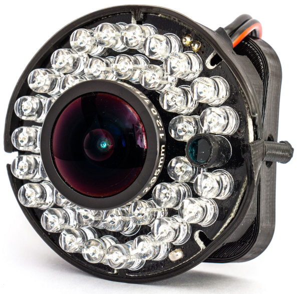


Figure 3.4: Specifications of the near-infrared camera package developed to record the Drive&Act dataset.

3.2 Evaluation of Camera Views for Cabin Monitoring

To capture the body pose of the driver it is necessary for the camera system to at least monitor the driver area of the vehicles interior. However, the activities we planned to include in our datasets also involved objects that could be located on the co-driver's side. In addition, future passenger monitoring systems may monitor the driver and co-driver with the same camera to reduce complexity and to save costs. Different sensor systems can add additional constraints, for example regarding a minimal distance to the driver or a limited field of view, as shown in the previous section. It is therefore necessary to identify suitable camera positions that satisfy these constraints.

We used two different depth sensors to record our datasets. Both were chosen based on the limited number of available depth cameras on the market. It was not possible to change their field of view nor the minimum distance requirement to generate depth data. Their final position in the vehicle was therefore largely determined by these constraints with the Melexis sensor mounted near the interior mirror and the Kinect mounted at the A-pillar on the co-driver side. Our experience with these sensors later helped with planning the multi-camera system used to record the Drive&Act dataset consisting of multiple near-infrared camera systems in addition to the Kinect TOF camera. The near-infrared camera system introduced no constraints compared to the depth cameras because of its interchangeable lenses and the small size of the camera package. To select a suitable position and field of view of each camera we conducted a small study by simulating camera data. The resulting images were evaluated subjectively with regards to the field of view in general and with regards to occlusion by the interior or by the passengers themselves. The overall goal was to position the cameras so the two front seats would be covered from different angles by the multi-view camera system while minimizing occlusions overall.

Figure 3.5 depicts all tested camera positions including their suitability to monitor the two front seats of the cabin. Rendered images for all views are shown in Appendix A. The

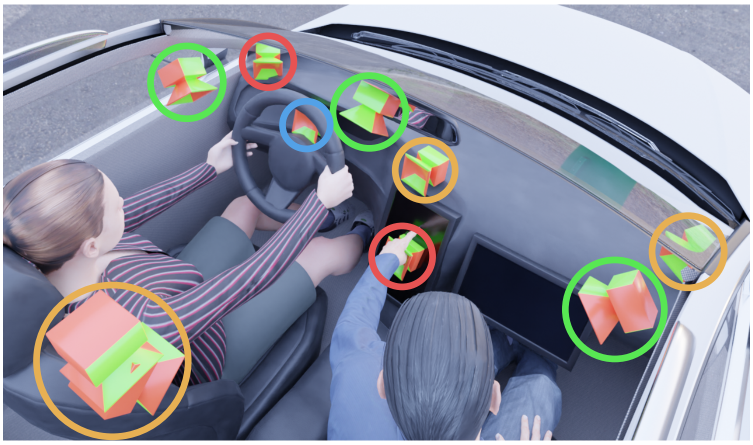


Figure 3.5: Overview of all camera positions evaluated on synthetic data rated from good (green) to unsuitable (red). The camera on the steering wheel column (blue) is only suitable to monitor the drivers head.

scene was modelled and rendered using Blender 2.8 [Com18]. 3D models of the passengers were generated using MakeHuman[5]. The reference 3D model of the car was an Audi A3 which was also the model used in the driving simulator of the Fraunhofer IOSB where the Drive&Act dataset was recorded (see Section 3.5.1). The evaluated scene depicted the driver in a typical driving posture. The co-driver interacted with the infotainment system which is a pose with larger displacement and a greater potential for occlusions. Positioning sensors in the interior is challenging because of the large windows. The remaining space is shared by structural elements, controls, airbags and other sensors. Despite these challenges, cameras need a good and unobstructed view. However, for most camera positions the legs are occluded by the steering wheel and are hidden in the foot well. They are therefore not considered by most driver monitoring systems including ours.

Overall, mounting positions above and in front of the passengers result in the best views with the least obstructions. The best positions are near the interior mirror as well as the top of the A-pillars. The view from the interior mirror is best suited to monitor both front passengers. However, the camera needs a large field of view of at least 120° horizontally. The view from the top of an A-pillar works well to monitor the passenger on the same side as the pillar. Unfortunately this position may result in interference when the passenger closer to the camera occludes the other passenger with their actions. The cameras need an equally large field of view of at least 120°. The ceiling of the car is also a position of interest, because it provides a good view of the dashboard. It is also the only view reaching into the foot well to capture the legs of the passengers. However, the face and, depending on the position of the seats, parts of the upper body of the passengers are occluded. Positions further down the A-pillars increase self-occlusion of passengers by

[5] http://www.makehumancommunity.org, accessed: June 14, 2022

raised arms and interference between driver and co-driver making them less suitable to monitor the whole cabin. Cameras at these positions are also further away from the passengers, decreasing the necessary field of view of the camera. The position is suitable on the co-driver side but not on the driver side because the steering wheel blocks large parts of the image. Cameras positioned on the steering wheel column are a popular choice for driver monitoring systems, but they cannot capture a larger area of the interior and are therefore not suitable for our experiments. Mounting a camera on the dashboard only offers limited visibility of the driver because of the steering wheel. In addition, the chance of the arms occluding large parts of the rest of the body increases. This problem gets even worse for a camera mounted lower in the infotainment area. Ribas et al. [Rib21] performed a similar experiment and came to similar conclusions.

According to this study we selected the three best views (i.e., interior mirror, top of A-pillars) as well as the ceiling camera to monitor the front of the cabin for the Drive&Act dataset. The Kinect was mounted on the co-driver A-pillar to monitor just the driver. This was the only option for this sensor because of its limited viewing angle. While we conducted this study using a model of an Audi A3, because this car was the recording environment for the Drive&Act dataset, the results should be applicable for other vehicles of similar design.

The suitability of specific sensor positions can change depending on the model of the car and the constraints of the sensor (e.g., field of view). The lower the angle of the front window and the farther it reaches back towards the heads of the passengers the more challenging it gets to position cameras in front of them because the required field of view increases. In bigger cars or buses this is less challenging. However, this means that similar studies should be conducted for each model to find the best sensor position. This also motivates our goal for sensor modality and sensor position invariant algorithms because it allows us to choose different positions and sensors for different cars without a large performance penalty or costly new data collections.

3.3 Driver Depth Pose Dataset

The Driver Depth Pose dataset was the first dataset collected for this thesis. It was created in cooperation with Robert Bosch GmbH where both data recording as well as annotation took place. For privacy reasons we can only show a small number of pictures of the collected data. The purpose of the dataset was to evaluate depth image-based 3D driver body pose estimation algorithms. To our knowledge there were no public automotive datasets for this task at that time. In the following sections we depict the data collection and annotation process in detail.

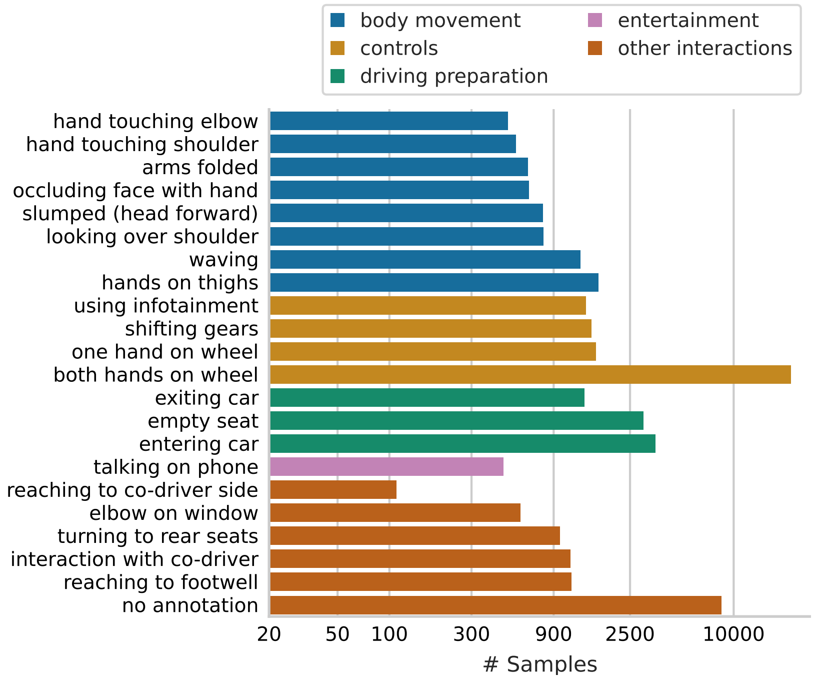


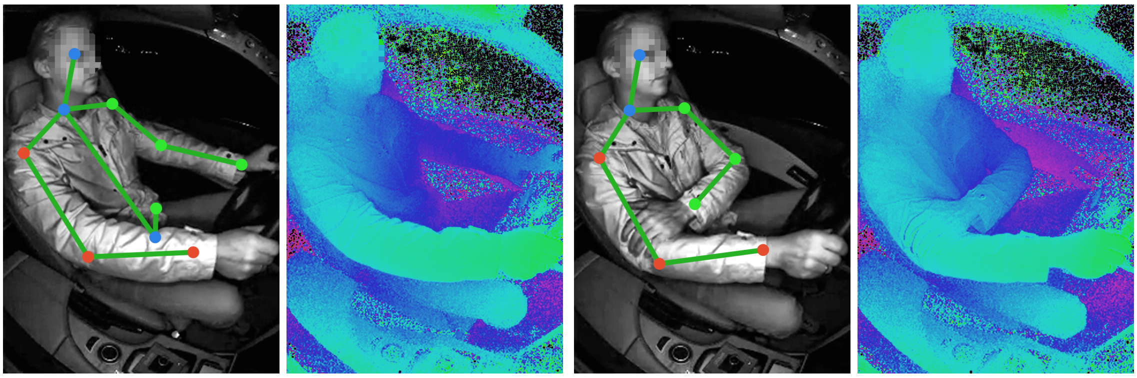
Figure 3.6: Movements and Actions performed in the Driver Depth Pose dataset together with their number of samples (logarithmic scale).

3.3.1 Data Collection

The goal of the dataset was to collect common body poses of drivers while driving in manual mode. Data collection took place in the laboratory but in the interior of a real car. The dataset was recorded with the automotive depth camera from Melexis as introduced in Figure 3.2. The sensor was mounted below the interior mirror facing the driver. As discussed in Section 3.2 this position minimizes self-occlusion of the driver. However, because of the limited field of view and resolution of the camera this only allowed us to monitor the driver's side of the cabin, which was sufficient for the proposed experiment. Both the depth image and the near-infrared image of the camera were recorded at 30 Hz.

Test participants received detailed instructions about what movements to perform and returned back to a normal driving posture with both hands on the steering wheel after each movement. Although the experiment was conducted in the cabin of a real car it was neither moving nor part of a simulation environment. The behavior of the test participants therefore reflected the instructed movements well, but it did not resemble the behavior while driving a real car. Overall, test participants were instructed to perform 21 different driving related movements and actions (see Figure 3.6).

The study was conducted with nine participants (eight male, one female). Five participants were recorded twice but wore a jacket for the second recording to change their body shape and increase the variance of the acquired data.



(a) Both hands on wheel

(b) Hand touching elbow

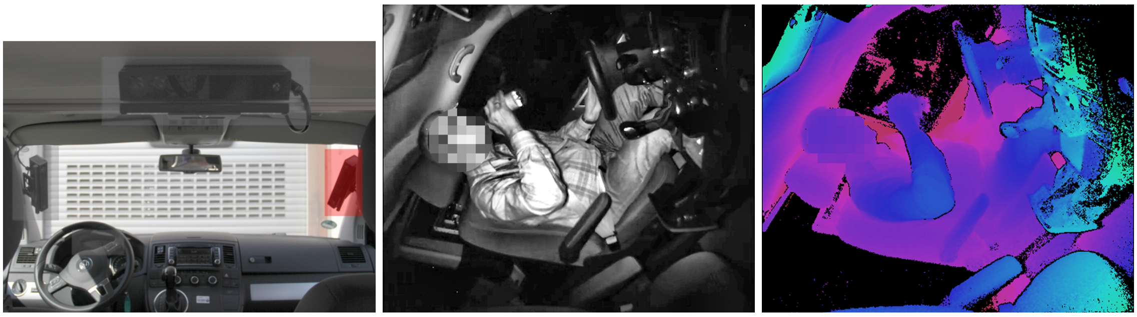
Figure 3.7: Sample images of the Driver Depth Pose dataset with annotated ground truth of eleven upper body keypoints (left) and corresponding depth image (right, false color)

3.3.2 Annotation

Overall, the dataset consists of 50 000 frames in 14 sequences with an average length of 2.5 minutes. Figure 3.6 shows how the recorded images are distributed across each activity. The action *both hands on wheel* is the most common by a large margin because it is the pose used in between each other instructed movement.

Each sequence was annotated manually with 11 keypoints (see Figure 3.7). The process involved two steps, first manual annotation of each 2D keypoint position in the near-infrared image followed by automated keypoint depth estimation via the corresponding depth image. To increase the efficiency of the manual annotation process linear interpolation was used to propagate annotations of past frames in the sequence to the current frame. The annotators then only needed to adapt incorrectly interpolated keypoints manually. In addition to the 2D keypoint positions annotators labeled if keypoints were occluded as well as the performed movement. Body poses in transition states could not be associated with any movement (i.e., *no annotation* in Figure 3.6).

Determining 3D positions using just the depth image posed some challenges which only allowed to label keypoints that were not occluded by looking up their depth value in the corresponding depth image. In addition, the resulting 3D keypoint positions were on the surface of the body instead of their true position inside the body. The 3D annotation followed the method presented in Equation 4.9. We depict the challenges regarding depth image-based 3D keypoint annotation in detail in Section 4.1. These drawbacks were alleviated in our public 3D driver body pose benchmark based on the Drive&Act dataset because it relied on a multi-view system for annotation instead of the depth camera (see Section 3.5.4).



(a) Camera Position [Son17b]

(b) Kinect NIR

(c) Kinect Depth

Figure 3.8: (a) Test vehicle interior with camera (red). (b, c) example images of the *drinking* activity for both modalities.

3.4 InCarIn Activity Dataset

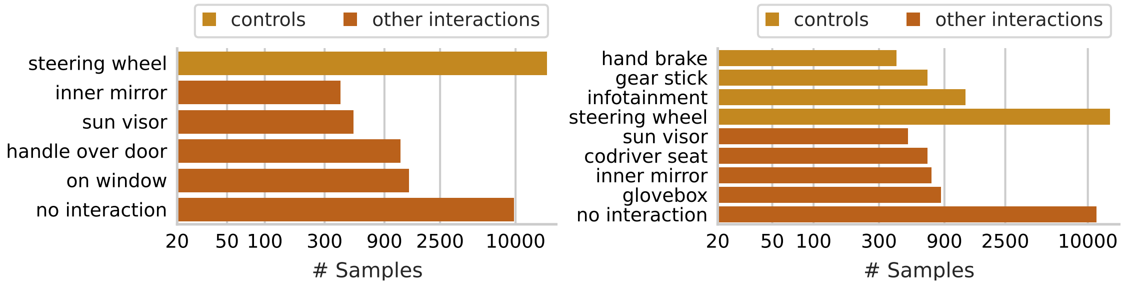
The InCarIn activity dataset was created in cooperation with partners of the BMBF funded project InCarIn[6]. It was recorded in the test vehicle of the project. The dataset was focused on activity recognition for manually driven vehicles and consisted of two parts. The first part was a pilot study to collect data of people interacting with the interior of the car, like the controls or infotainment area. The main dataset then focused on distracting secondary activities while driving the test vehicle in manual mode on a test track.

In the following we describe the hardware setup and both studies in detail.

3.4.1 Recording Environment

The dataset was collected in a Volkswagen T5 Multivan, which was the test vehicle of the project. Based on our evaluation of different sensor technologies we chose the Kinect 2 for Windows (see Figure 3.3) to record the data. Because of its limited viewing angle and the size of the camera it was mounted along the A-pillar on the co-driver side to be able to capture the whole driver side of the vehicle. Compared to the camera position used to record the Driver Depth Pose dataset this resulted in a more challenging side view with more occlusions (see Figure 3.8). This is also in line with our evaluation of different camera positions for the Drive&Act dataset as discussed at the start of the chapter. To collect the datasets the near-infrared video stream and the depth video stream of the Kinect were recorded at 30 Hz.

[6] https://www.interaktive-technologien.de/projekte/incarin, accessed: June 15, 2022



(a) Left Hand

(b) Right Hand

Figure 3.9: Number of samples per class of the InCarIn Interior Interaction dataset for the left and right hand (log sale).

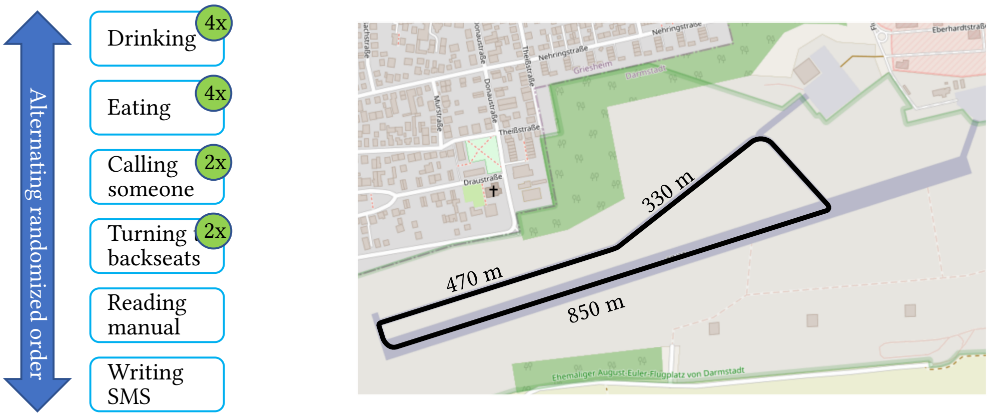
3.4.2 Pilot Study - InCarIn Interior Interaction Dataset

We assumed that knowing the accurate position of the hands within the vehicle's interior, based on 3D driver body pose estimation, would be a good cue to determine what parts of the interior passengers interact with. To test this hypothesis, we recorded and annotated a small dataset for vehicle interaction detection. In addition, this collection served as the first test of the recording environment prior to the more extensive collection of secondary activities on a test track.

Data was recorded in the stationary vehicle. We instructed five people (all male) to reach for ten regions within the car's interior. This was repeated multiple times with both hands returning to a driving posture, with hands on the steering wheel, after each interaction. The data was annotated manually afterwards making sure to only label frames as interacting with an element if the element was either touched (i.e., in case of the infotainment area) or grabbed (i.e., in case of the steering wheel or gear lever). There were two labels per image corresponding to interaction with the left and the right hand.

Overall, the dataset consisted of 31 000 frames labeled as 11 classes (10 interaction targets and the class no interaction). In contrast to the activity recognition datasets presented in the following each sample of this dataset was just one frame instead of a video segment. Figure 3.9 shows the statistics of the dataset as well as the instructed interaction areas. The two most common classes are *steering wheel* and *no interaction*. This follows expectations because grabbing the steering wheel was the instructed position between other interactions. The class *no interaction*, on the other hand, included transition phases as well as resting positions where participants put their hands for example in their lap. There are fewer interaction zones for the left hand because it is more confined by the driver door and most controls can only be reached comfortably with the right hand.

We demonstrate the usefulness of the 3D driver body pose for interior interaction detection in Section 5.1 verifying the initial assumption of this section.



(a) Experiment protocol. **(b)** Griesheim airfield test track[7].

Figure 3.10: InCarIn activity dataset experiment protocol (a) and test track (b).

3.4.3 Data Collection

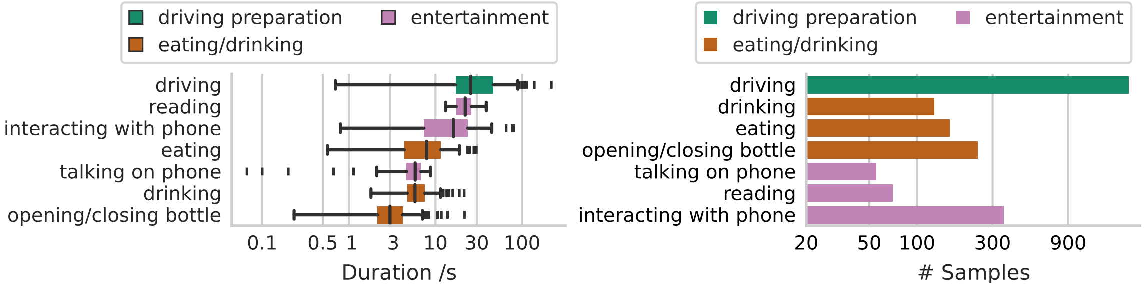
Following the successful collection of the *InCarIn Interior Interaction* dataset the main goal was to collect data of distracting secondary activities while driving manually. The study was conducted in close cooperation with the Fraunhofer IAO in Stuttgart.

The secondary activities were selected by talking to experts of project partners as well as by a user study conducted by project partners. The resulting list of actions was checked for feasibility and reduced to a number that could reasonably be recorded in the planned experiment with enough variance and samples. Overall, we selected five secondary activities. Figure 3.10a shows the test protocol of the study including the selected secondary activities. To increase the number of samples short activities like drinking and eating were repeated multiple times. The order of all activities was randomized while making sure to alternate between different activities. Test participants also had to turn and talk to people on the back seat. This was less needed for the activity recognition dataset but was used by project partners to test an eye tracking system. It nevertheless served to break up repeats of the same activity. Secondary activities, according to the protocol, were instructed by an examiner seated in one of the back seats. The participant had to perform the action as soon as safely possible. Longer activities (e.g., reading) sometimes had to be interrupted to turn corners on the track.

The selected activities were highly distracting and in part forbidden in normal traffic, the experiment therefore took place on a closed-off airfield (see Figure 3.10b). On average test participants drove between 30 to 50 km/h.

Overall, we collected data of 26 test subjects (10 male, 16 female).

[7] Map Data © OpenStreetMap contributors (https://www.openstreetmap.org/copyright)



(a) Activity Duration

(b) Sample Frequency

Figure 3.11: The average duration of each activity (a) as well as the resulting number of samples per class (b) of the InCarIn activity dataset (logarithmic scale).

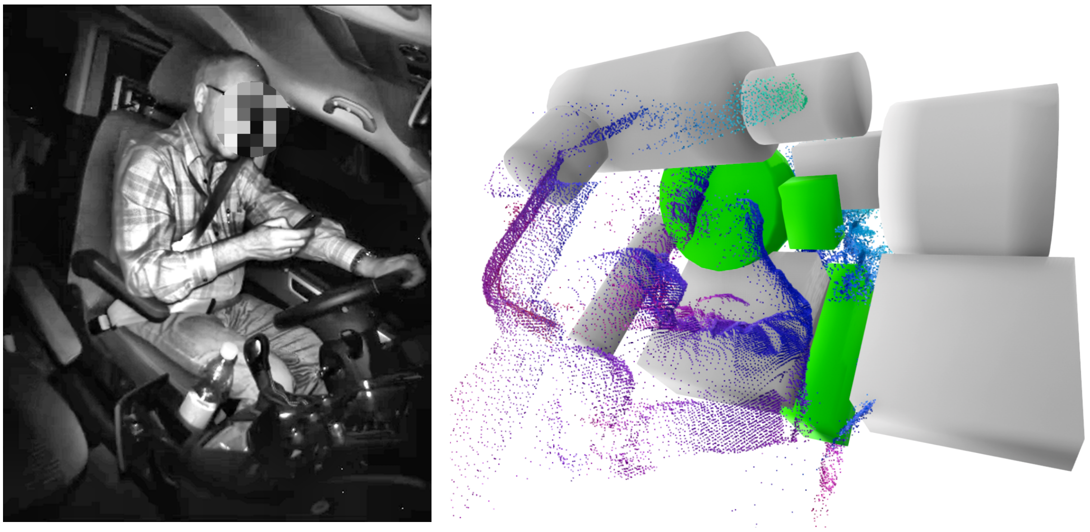
3.4.4 Activity Annotation

The data collection resulted in 26 video streams including two image modalities (Depth and NIR). On average each sequence took 13 minutes, resulting in overall 5:45 hours of video data. After analyzing the data, we decided to split the action *drinking* into two parts *opening/closing a bottle* and *drinking*. This resulted in seven activity classes (six secondary activities and the primary driving task). All videos were manually labeled, marking the start and end frame of each activity. Figure 3.11a shows duration statistics of the labeled intervals. Unsurprisingly the intervals for driving the vehicle were the longest. Overall, 67 % of the dataset were labeled as *driving*. In addition, the average duration of the secondary tasks varied greatly, with *opening/closing a bottle* and *reading* taking 80 and 700 frames on average, respectively.

As shown in our review of related work, activity recognition datasets usually consisted of short snippets instead of long sequences. However, the intervals of our data were very uneven and in part much longer compared to other activity recognition datasets. To provide a similar setup and to increase the number of samples per class we split all intervals into sections of 300 frames or less. Figure 3.11b shows the number of samples per label after creating these chunks. The samples were not equally distributed across classes. *Driving* is the most common class by a large margin. However, all secondary activities were still represented by at least 50 samples.

3.4.5 Interior Annotation

One of the research goals of this thesis was the investigation of additional input modalities for 3D human body pose based activity recognition. The classes labeled in the pilot study for interior interaction detection naturally correspond to different areas of the vehicle. However, even secondary activities like *drinking* require the driver to pick up the drink from some storage area, like the co-driver seat or cup holder. Labeling the interior was



(a) NIR image

(b) Point cloud with interior annotations

Figure 3.12: InCarIn dataset sample for activity *interacting with phone*. (a) NIR image of the Kinect (b) Point cloud with manually annotated interior elements. Controls are colored in green.

therefore a natural extension for driver body pose based activity recognition systems. We therefore generated a static model of the interior containing relevant areas for both interior interaction classes as well as secondary activities.

A common way to describe 3D surfaces are mesh models. However, creating such a mesh is difficult as it either requires 3D construction or plans from the car manufacturer. We instead decided to represent the interior with 3D primitives like cubes and cylinders. To position these primitives manually we utilized the point cloud generated from the recorded depth images and placed primitives with suitable shapes at the areas of the point cloud corresponding to certain interior elements. The steering wheel, for example, was easily visible in the point cloud and could accurately be represented by a cylinder. This approach had the added benefit of not only describing the current surface of the interior but also areas where things would move. For example, the sun visor, when deployed, rotates around a hinge. This area was therefore represented as a cylinder in our model covering the volume that would be part of the interaction with the area.

Figure 3.12 shows the point cloud used for annotation as well as the annotated 3D interior elements. Overall, the interior consisted of eleven primitives including the ten areas that were part of the pilot study for interior interaction detection and in addition the seat of the driver.

3.4.6 Summary

The InCarIn activity dataset was our first dataset for driver activity recognition based on depth data. Compared to related automotive datasets it was, to our knowledge, the first to include depth data as well as the first with a size of more than one million frames. However, while we contributed methods using this data, we could not make the dataset itself publicly available for privacy reasons. The dataset consisted of two parts:

The Interior Interaction benchmark included 31 000 frames manually labeled with eleven classes. It focused on interactions of the driver with interior elements, like the steering wheel. These activities were therefore directly related to specific locations in the interior. The complete data was only used for testing as our method did not rely on machine learning. It was also the only activity benchmark collected by us that was evaluated frame by frame and was not based on time series data.

The Activity dataset included 1.2 million frames in 26 streams recorded from 26 people driving manually on a test track. The data was manually labeled with seven activity classes followed by the extraction of samples with a length of 300 frames or less. To test activity recognition methods with this dataset we created a cross-person benchmark using data of fifteen people for training, two for validation and nine for testing.

Apart from the annotation of activities we also labeled interior elements, like the steering wheel, using eleven geometric primitives. The interior model was our first addition to the input modalities of 3D driver body pose based activity recognition methods. We could show its benefit for both parts of the dataset (see Section 5.1 and Section 5.2).

3.5 Drive&Act Dataset

The Drive&Act dataset concluded the data collection efforts of this thesis. It combined the knowledge gained by recording the previous datasets, alleviated most of their drawbacks and extended the domain to automated driving. It was the primary dataset of this thesis and was the benchmark for all algorithms both for driver body pose estimation as well as activity recognition.

The greatest drawback of the Driver Depth Pose dataset was its simple study design, involving just the driver, and the limited number of instructed poses. In addition, the camera setup with just a single sensor was limiting with regards to 3D driver body pose annotation. Drive&Act on the other hand included a cluttered car interior, unconstrained natural movement and a complex multi-view and multi-modal camera setup. It enabled the generation of superior ground truth data for 3D driver body pose evaluation including body parts occluded in some views.

While the InCarIn driver activity dataset was larger in size compared to related driver activity datasets of that time, it was still smaller and less complex than datasets from other domains. Its greatest drawback was its small number of secondary activities. However, collecting more varied secondary activities with a focus on automated driving was not possible in a manually driven car because of safety concerns. The Drive&Act dataset was therefore recorded in a driving simulator. While this did not allow us to capture environmental influences like vibrations or illumination changes, it enabled us to include many distracting activities that will likely occur in future automated cars. Drive&Act therefore vastly expanded both the number of performed activities as well as their complexity.

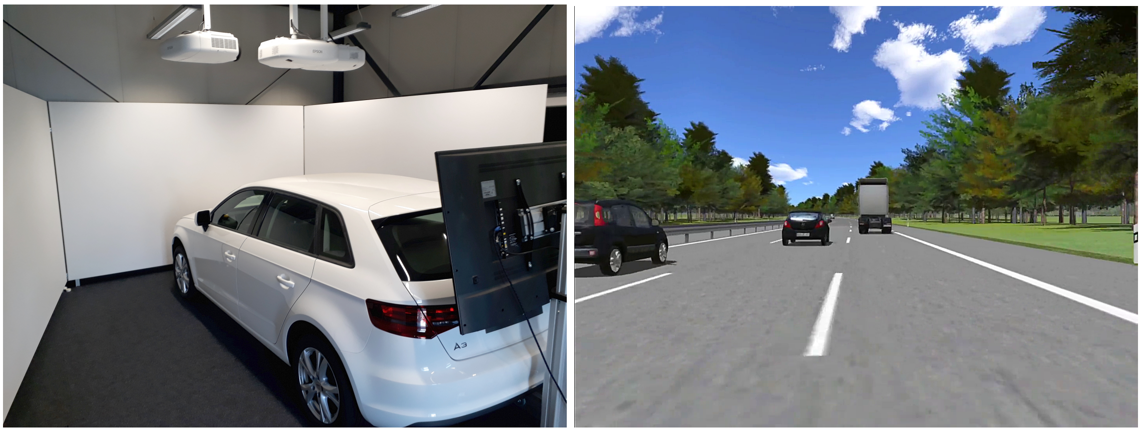
The dataset was recorded in cooperation with project partners in the BMBF funded project PAKoS[8]. The resulting dataset was published at the International Conference on Computer Vision [Mar19] © IEEE, 2019. The following chapter is based on this publication. Alina Roitberg and Manuel Martin contributed equally to the dataset. It is part of both of their theses. Manuel Martin was responsible for the implementation and experiments regarding the body pose-based approaches (which are described in depth in Chapter 5), while Alina Roitberg implemented the video-based models. Their results are provided for comparison only. With regards to the creation of the Drive&Act dataset, both have contributed significantly to all phases of data collection as part of their PhD research. While setting a strict line is hard, Alina Roitberg focused more on the annotation of the activities as a hierarchy, while Manuel Martin focused on the sensor and simulator setup. The accents of this thesis were set accordingly. This chapter also discusses extensions of Drive&Act with additional object bounding box and 3D position annotations as well as with a benchmark for 3D driver body pose estimation. Both extensions were published separately and were the sole contribution of Manuel Martin [Mar20a, Mar21].

3.5.1 Recording Environment

The recording environments between the presented datasets varied widely. The Driver Depth Pose dataset was recorded in the least realistic and immersive way in a laboratory but without simulating the driving task. The InCarIn Activity dataset on the other hand was recorded in a realistic way on a test track driving in manual mode. However, this restricted what secondary activities could be performed because the driver still had to drive safely.

For the recording of the Drive&Act dataset we chose a middle ground between both cases. Data was recorded in a real car but within a static driving simulator (see Figure 3.13). The vehicle surroundings were simulated and projected on multiple screens around a modified

[8] https://www.interaktive-technologien.de/projekte/pakos, accessed: June 15, 2022



(a) Simulator environment

(b) Simulation on center screen

Figure 3.13: Overview of the simulator hardware (a) and simulation with SILAB (b).

Audi A3 using the simulation software SILAB[9]. The simulation spanned a field of view of 200° in front of the driver and included additional screens for both exterior mirrors and the interior mirror. The driver could interact with the simulation using the normal controls of the car. Forces on the steering wheel and pedals were simulated but vibrations and vehicle dynamics were not. Manual driving was supported by the simulation software. The automated mode was specifically developed to collect the Drive&Act dataset, including take-over requests by the automation. The simulation served as a background for the collection of the dataset to keep test participants engaged and to create the experience of driving in an automated car with SAE Level 3 capabilities. The test track consisted of a straight and endless piece of highway with moderate traffic. The design of the track decreased the risk of simulator sickness because it did not contain any turns or curves. In addition, because the track was endless the duration of the data collection experiment was not limited by the simulated scenario.

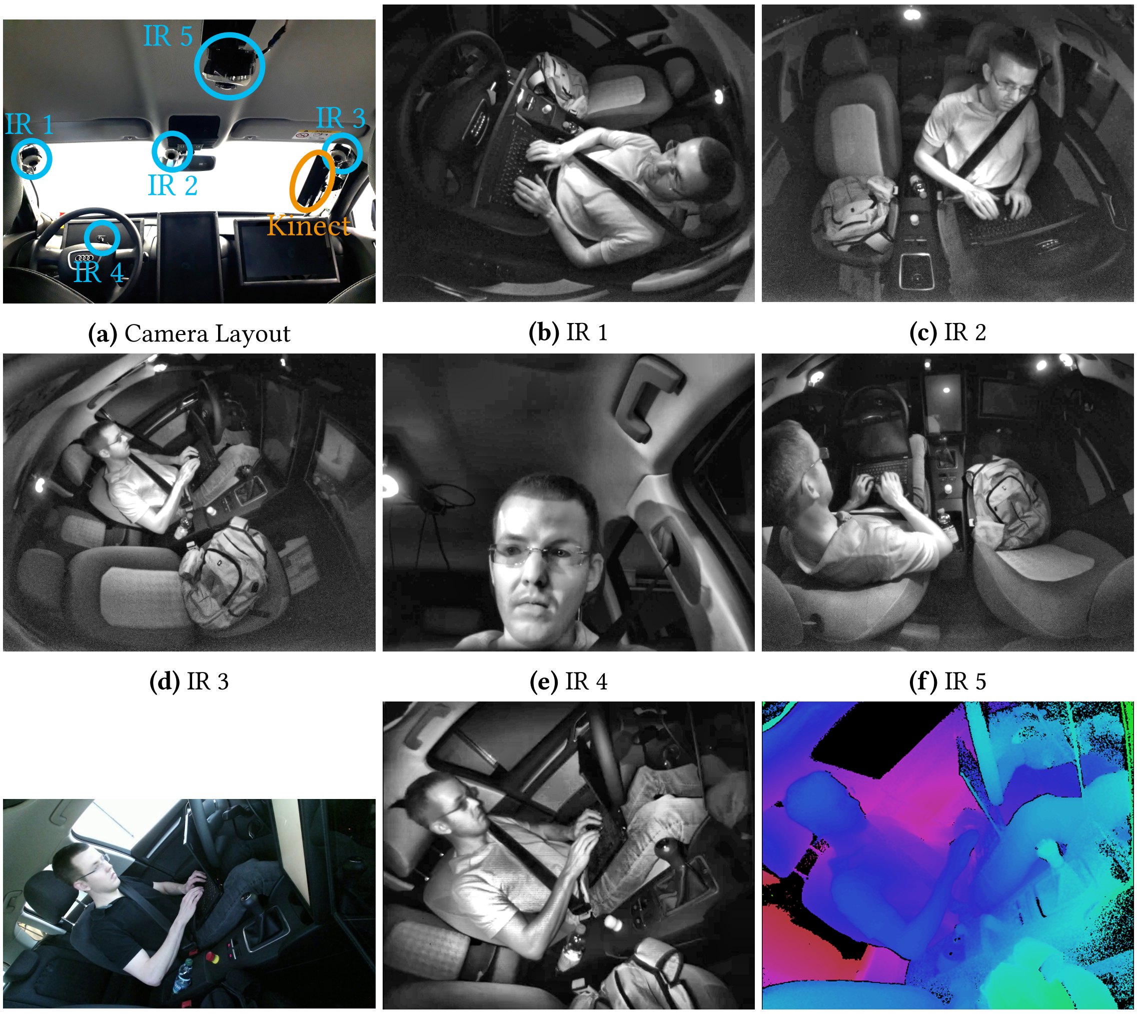
Figure 3.14a shows the interior of the vehicle. It was heavily modified with mounting points for cameras. The figure also shows the camera setup for the recording of the dataset. While the previous datasets were recorded with a single depth camera, the setup for Drive&Act was comprised of six cameras, one Kinect 2 for Windows (see Figure 3.3 for specifications) and five NIR camera systems specifically developed to record the dataset (see Figure 3.2 for specifications). The Kinect was positioned at the co-driver A-pillar, same as for the InCarIn Activity dataset. Four NIR camera systems monitored the cabin, and one monitored the driver's head. Overall, this placement covered most camera perspectives identified in our evaluation of suitable camera positions for cabin monitoring (see Section 3.2). All cameras were calibrated, using a checkerboard pattern, and formed

[9] https://wivw.de/en/silab, accessed: June 15, 2022

a multi-view system including the depth sensor and its point cloud. Figure 3.14 shows all views recorded by the system.

The recording of both previous datasets was guided by a supervisor who instructed the test participants. The recording of the Drive&Act dataset was instead self-guided. To achieve this, the instructions were displayed on the central touch screen of the simulator. The process is detailed in the next section.

All data was recorded timestamp synchronized using the Robot Operating System (ROS) [Qui09]. This included the camera streams (at 30 Hz), simulation parameters as well as interactions of the test participants with the instruction system on the central screen.



(a) Camera Layout **(b)** IR 1 **(c)** IR 2

(d) IR 3 **(e)** IR 4 **(f)** IR 5

(g) Kinect Color **(h)** Kinect NIR **(i)** Kinect Depth

Figure 3.14: Example images of the *working on laptop* activity for different views and modalities.

3.5.2 Data Collection

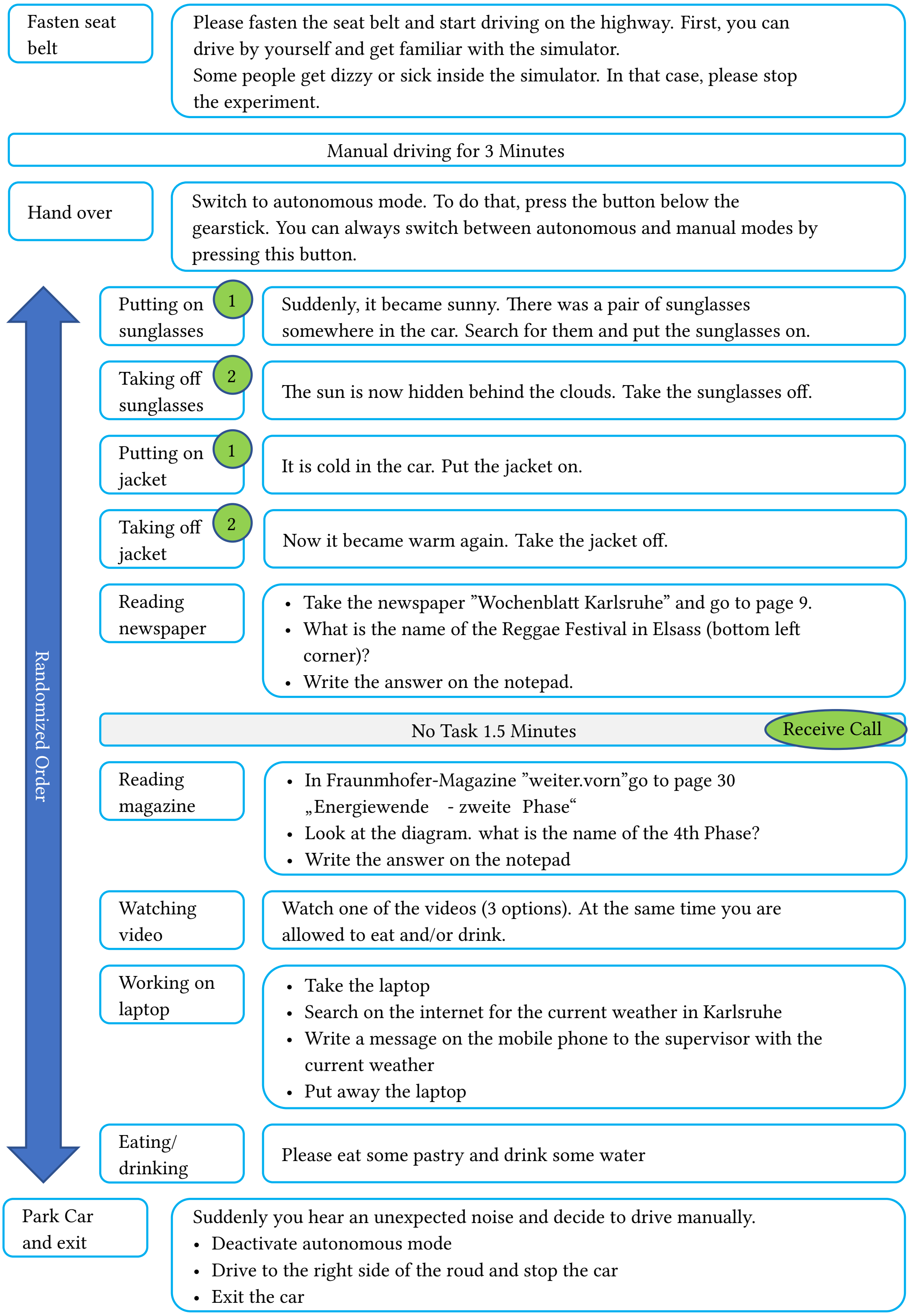
The goal for collecting the Drive&Act dataset was to create a dataset for activity recognition that would surpass all available public sources for driver monitoring at that time and would close the gap to general activity recognition datasets. In addition, it should advance driver monitoring for automated vehicles. Like the InCarIn activity dataset, activities for the Drive&Act dataset were identified by project partners via literature research as well as expert interviews. However, collecting data in the simulator left us more freedom to increase the number of activities as well as their complexity. The recorded activities are detailed in the next section.

Other driver monitoring datasets, including our previous datasets, were collected by instructing test participants to perform specific activities. While this is an efficient way to collect data, it limits the realism of the recorded data with regards to the duration of each task and their order. We therefore used a self-paced approach to record the Drive&Act dataset by grouping activities into complex tasks. Test participants worked through each task performing each activity to reach the goal at their own pace and in their own order. In addition, before each recording session, fifteen objects required for the tasks were randomly distributed in the car without the knowledge of the test participants. This further randomized the execution of each activity because objects had to be found and picked up from various locations. Overall, test participants had to complete twelve tasks instructed on the central touch screen.

Preparations before the experiment were minimal consisting of a short questionnaire about demographics, driving experience and experience of assistive functions, a consent form including the publication of the data, and general information about the simulator and the possibility of simulator sickness. With regards to the experiment itself, we instructed test participants to follow the instructions presented on the center screen. In addition, we noted that they were free to engage in additional activities not part of the current task, like eating, drinking, or checking their phone. Some participants made heavy use of this, increasing the duration and variance of the recording session, while others strictly followed the instructions on the center screen. The goal of these instructions was to provoke as much variation and natural behavior as possible while making sure to collect data of all planned activities.

Figure 3.15 depicts the protocol of each data collection session including the twelve tasks and the instructions provided to the test participants on the center screen. Test participants did not know the protocol and were only presented with a single task at a time. The experiment started with the first task to *fasten the seat belt*. It was followed by a brief period of manual driving to get familiar with the simulator and its handling. Afterwards participants were presented with the next task to *hand over* the driving task to the automation. The order of the following tasks was randomized keeping specific activities in order, if necessary, like *putting on sunglasses* before the activity to take them off again. After each task there was a short, randomized pause of at least ten seconds to give the participants the opportunity to check their surroundings including the simulation or to engage in other activities. One of these pauses was programmed to be longer. The instructor used this time to call the participant on the phone located somewhere in the vehicle. After the completion of all randomized tasks, the last task instructed the participant to *park the car and exit.* With just this protocol test participants could disengage completely from the driving task because they would not be involved for most of the time. To prevent this, the instructor triggered four unexpected take-over requests in each session interrupting the running task. Test participants had to take over and drive manually for one minute before finishing their task.

Fifteen people, four female and eleven male, participated in the data collection. We selected participants of different body height and weight, as well as different driving styles and familiarity with assistance systems and automation modes. Most participants were recorded twice, resulting in 29 recorded driving sessions with an average duration of 24 minutes. Most participants took less time during the second session, as they were familiar with the tasks, resulting in overall different behavior and more variety in our dataset.



Fasten seat belt

Please fasten the seat belt and start driving on the highway. First, you can drive by yourself and get familiar with the simulator.
Some people get dizzy or sick inside the simulator. In that case, please stop the experiment.

Manual driving for 3 Minutes

Hand over

Switch to autonomous mode. To do that, press the button below the gearstick. You can always switch between autonomous and manual modes by pressing this button.

Putting on sunglasses

Suddenly, it became sunny. There was a pair of sunglasses somewhere in the car. Search for them and put the sunglasses on.

Taking off sunglasses

The sun is now hidden behind the clouds. Take the sunglasses off.

Putting on jacket

It is cold in the car. Put the jacket on.

Taking off jacket

Now it became warm again. Take the jacket off.

Reading newspaper

- Take the newspaper ”Wochenblatt Karlsruhe” and go to page 9.
- What is the name of the Reggae Festival in Elsass (bottom left corner)?
- Write the answer on the notepad.

No Task 1.5 Minutes

Reading magazine

- In Fraunmhofer-Magazine ”weiter.vorn”go to page 30 „Energiewende - zweite Phase“
- Look at the diagram. what is the name of the 4th Phase?
- Write the answer on the notepad

Watching video

Watch one of the videos (3 options). At the same time you are allowed to eat and/or drink.

Working on laptop

- Take the laptop
- Search on the internet for the current weather in Karlsruhe
- Write a message on the mobile phone to the supervisor with the current weather
- Put away the laptop

Eating/ drinking

Please eat some pastry and drink some water

Park Car and exit

Suddenly you hear an unexpected noise and decide to drive manually.
- Deactivate autonomous mode
- Drive to the right side of the roud and stop the car
- Exit the car

Figure 3.15: Drive&Act experiment protocol showing task name (left) and subject instructions (right). Numbered tasks were randomized but kept in rising order.

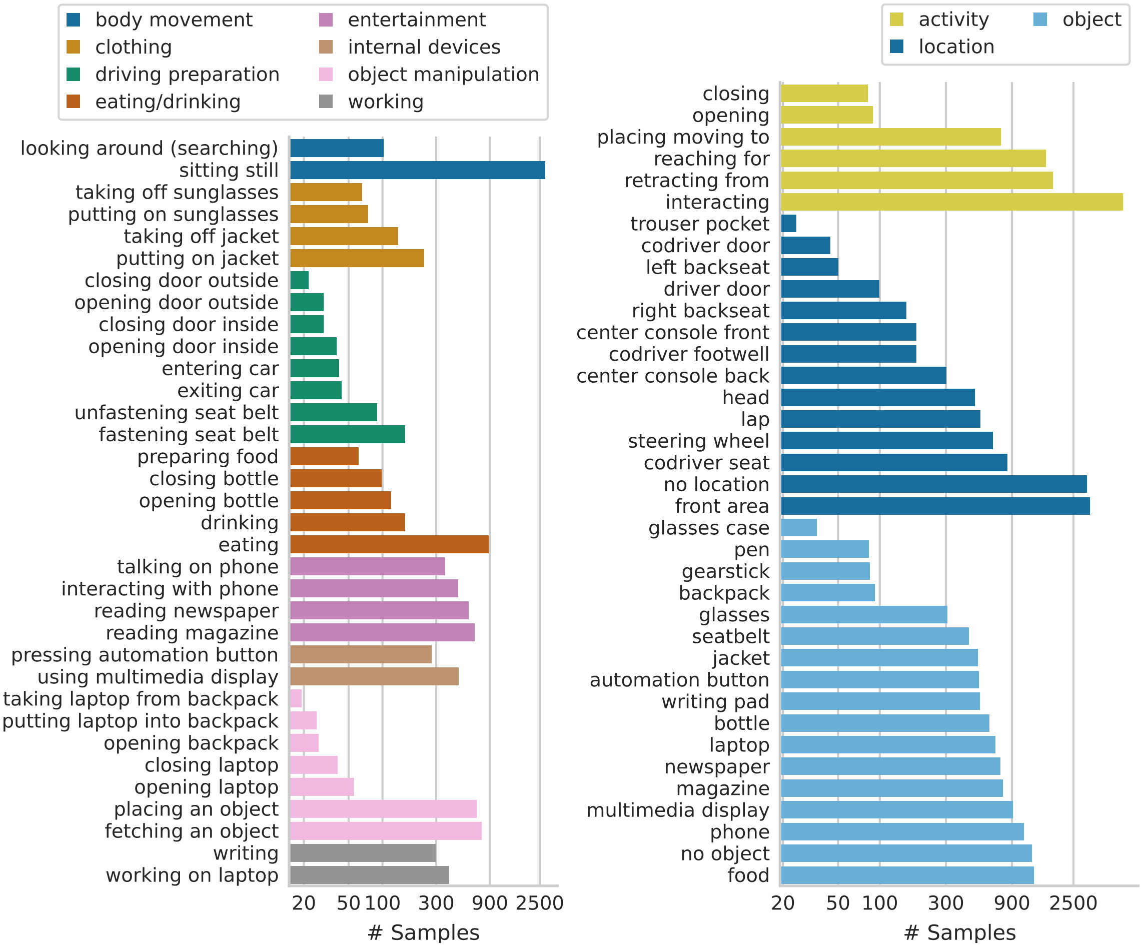
3.5.3 Activity Annotation

The annotation process was in large parts the same as for the InCarIn dataset. The recorded video frames were manually labeled by marking the start and end of each activity in the video. The annotated segments varied widely in length. We therefore split each segment into three second chunks that formed the samples discussed in the remainder of this section. The annotations were hierarchical in three levels of abstraction, resulting in 83 activity classes in total. The hierarchy targeted high-level scenarios, fine-grained activities, which retain semantic meaning, and low-level atomic action units, which represent environment and object interactions. The fine-grained activities were based on the results of the literature research as well as expert interviews by project partners. They were refined after the data collection by analyzing the recorded video data, for example by splitting activities like drinking into multiple parts. Fine-grained activities were aggregated to form the tasks the test participants had to solve, and they were further decomposed to form the low-level atomic action units:

Scenarios / Tasks: The twelve tasks our subjects had to complete in each session (see Figure 3.15) shaped the *first level* of our hierarchy and were either scenarios typical during manual driving (e.g., *eating and drinking*) or highly distracting situations which are expected to become common with increasing automation (e.g., *using a laptop*). In general, this level represented composite and long-lasting activities. Subjects spent most of the time (23 %) in the entertainment task (i.e., watching a video), and the shortest time driving manually after a take-over request. The *take over* scenario was special because the subject was unexpectedly asked to interrupt what he was doing to take over and switch to manual driving.

Fine-grained Activities: The *second level* represents *fine-grained activities*, breaking down the *scenarios / tasks* into 34 concise categories. In contrast to the upcoming third level of *atomic action units*, the second level classes preserved a clear semantic meaning. These fine-grained activities alternated freely during a scenario i.e., the driver was *not* told *how* to execute the task in detail. Of course, there was a strong causal link between different degrees of abstraction, as composite behaviors often were comprised of multiple simpler actions.

A key challenge for recognition at this level was the concise nature of the classes, as we differentiated between *closing bottle* and *opening bottle* or between *eating* and *preparing food*. We argue that such detailed discrimination is important for applications, as the coarse components of the scene (i.e., the vehicle cabin or the loose body position) often remain similar and the relevant class-differences occur at a smaller scale than in traditional action recognition benchmarks. As a consequence of such detailed annotation the frequency of individual classes varied (see Figure 3.16a). On average, the



(a) Fine-grained Activities

(b) Atomic Action Units

Figure 3.16: Sample frequency of composite activities (a) and atomic action units (b) by class (logarithmic scale). A sample corresponds to a three second snippet with assigned label. Colors denote the activity group (e.g., food-related activities).

dataset featured 303 samples per class, with *taking laptop from backpack* being the least represented (19 samples) and *sitting still* being the most frequent category (2797 samples).

Atomic Action Units: The annotations of *atomic action units* portrayed the lowest degree of abstraction and were basic driver interactions with the environment. The action units were detached from long-term semantic meaning and could be viewed as building blocks for complex activities of the previous levels. We defined an atomic action unit as a triplet of *action*, *object* and *location*. We covered 5 types of actions (e.g., *reaching for*), 17 object classes (e.g., *writing pad*) and 14 location annotations (e.g., *co-driver footwell*), with their distribution summarized in Figure 3.16b. Overall, 372 combinations of action, object and location were captured in our dataset.

3.5.4 3D Driver Body Pose Annotation

The body pose of the driver was the most important input feature of our activity recognition experiments. We provided automated annotations for the 3D upper body pose of the driver with 13 keypoints as part of the published dataset for activity recognition. To generate this annotation, we used *OpenPose* [Cao18] for 2D driver body pose estimation. We obtained the 3D body pose of the driver via triangulation of 2D poses from three frontal views (both cameras at the A-pillars and the camera at the central mirror). The evaluation of our activity recognition methods on Drive&Act also relied on this annotation to be comparable to others using the public data. However, in addition we also tested the performance of our methods using the depth image-based 3D driver body pose approaches contributed by us.

Research on 3D driver body pose estimation was an integral part of our work. We introduce our methods based on depth cameras as well as the triangulation-based method used for the public annotations in Chapter 4. However, to evaluate these methods suitable annotated data was necessary. While we already presented the Driver Depth Pose dataset, it only included data from a single depth camera which made it unsuitable to evaluate triangulation-based methods. In addition, the single sensor design of the Driver Depth Pose dataset proved challenging for accurate ground truth labeling because the depth camera could only observe the surface of the human body from a single viewpoint (see Section 4.1).

To alleviate both problems we created an additional 3D driver body pose benchmark based on the Drive&Act data. The most accurate way to annotate 3D driver body poses would have been marker-based motion capture. However, as depicted by Borges et al. [Bor20] the confined space in the interior would have made this difficult. When we were recording the Drive&Act dataset, ground truth for 3D driver pose estimation was not the focus of our efforts. Due to the multi-view setup of the dataset, we were able manually annotate 2D driver body poses in multiple views followed by triangulation of these annotations to generate 3D ground truth. This was the same process used to generate the automated 3D driver body pose labels provided for the whole dataset but with manually labeled 2D poses as a starting point. We chose the same three NIR cameras as for the automated annotation in addition to the NIR image of the Kinect and labeled the 2D driver body pose using the annotation tool CVAT in all four views. Afterwards the data was triangulated and re-projected to each view. The re-projected 2D driver body poses were then manually checked again to fix any annotation errors. The triangulation of the corrected 2D driver body poses then served as ground truth for our experiments.

Drive&Act consists of many frames annotating even a fraction of the data manually would have been very time consuming. Selecting frames at random would not ensure the uniqueness of each body pose. We therefore used the automated annotations generated with *OpenPose* to guide the selection process. We sampled frames equally distributed from all fine-grained activities. A sample was included in the body pose benchmark if the distance of the candidate 3D driver body pose differed by at least 10 cm in one body keypoint from all previously selected samples. This process ensured that all activities were represented in the body pose estimation benchmark and that all included 3D body poses were unique.

We applied this approach to the data of four participants of the Drive&Act dataset to extract a 3D driver body pose benchmark consisting of 1500 highly challenging annotated 3D scenes depicting different activities, including occlusions. This corresponded to 6000 manually labeled 2D driver body poses annotated across four views that could also be used to evaluate 2D driver body pose detectors. The manually labeled data was strictly used for testing because of its limited size. However, the automated annotations provided with the dataset of the eleven remaining participants could be used for training purposes if needed. We also tested the accuracy of these automatically created labels on the new manual annotations (see Section 4.4.3). The 3D driver body pose benchmark was made public as part of our publication at the International Conference on Intelligent Transportation systems (ITSC) in 2021 [Mar21].

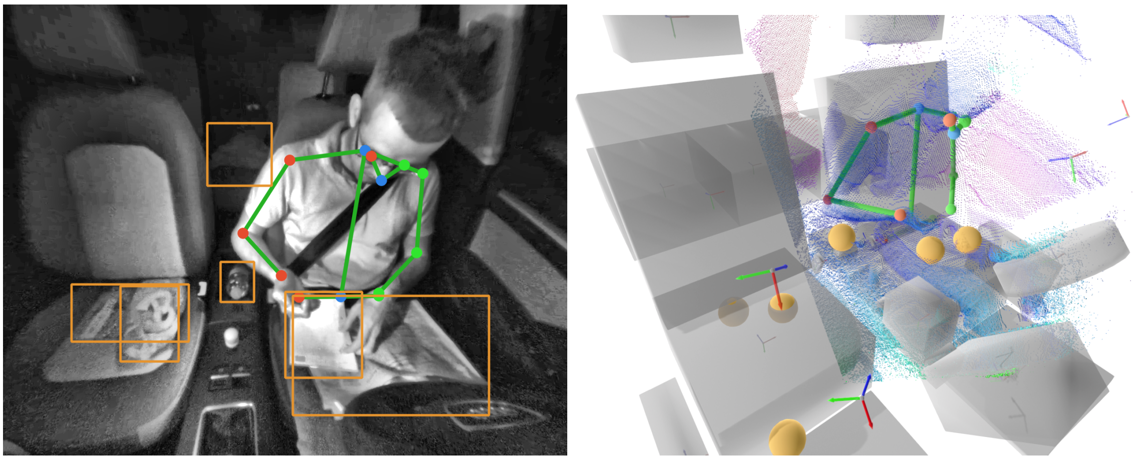
3.5.5 Interior Annotation

We succeessfully demonstrated the benefits of using a model of the interior for activity recognition with the InCarIn dataset. This is also why we created a similar model consisting of geometric primitives to mark controls, seats, and other areas relevant for the activities of the Drive&Act dataset. The annotation process was explained in Section 3.4.5.

While the resulting interior model included similar primitives for the controls of the car, like the steering wheel or gear lever, other regions differed. Some areas only relevant for the interior interaction task of the InCarIn dataset were missing in Drive&Act (e.g., sun visor) while others were represented with more detail like the seats, including the back seats, as well as storage areas relevant for Drive&Act activities. Figure 3.17 depicts the resulting interior model. Overall, it consisted of 28 geometric primitives.

3.5.6 Bounding Box Annotation

To continue our efforts extending the input modalities of 3D human body pose based activity recognition methods even beyond the 3D interior model, we also annotated bounding boxes and 3D tracks of the most important objects used in Drive&Act. We experimented with object detectors but because of severe occlusions and hard to detect object



(a) 2D annotations

(b) 3D annotations

Figure 3.17: Manual annotations for the Drive&Act dataset including bounding boxes and 3D positions of objects (orange), interior elements (gray) and the 2D and 3D body pose of the driver (green).

classes (e.g., newspapers are highly deformable, smartphones are often occluded) we decided to manually annotate object bounding boxes. Overall, we annotated nine objects (bottles, phone, magazine, newspaper, backpack, food, writing pad, jacket, laptop). These annotations were made public as part of our publication at the International Conference on Intelligent Transportation Systems (ITSC) in 2020 [Mar20a].

The annotation process involved two steps. The first step was manual annotation of bounding box tracks for each object in four camera views, including the NIR Image of the Kinect, and the NIR Cameras at the A-pillars and at the interior mirror. These were the same views used to generate the 3D driver body pose benchmark. We used CVAT[10] to annotate the bounding box tracks making use of interpolation between frames to create labels for the whole dataset. The resulting bounding box tracks describe the movement of each object in the image of each annotated camera view.

The second annotation step was to create 3D trajectories for each object to be able to integrate them both with the 3D interior model as well as the 3D body pose of the driver for our activity recognition experiments. Figure 3.17 shows a sample of the annotated bounding boxes and the resulting 3D object positions. We planned to compare NIR camera based multi-view systems with using just the Kinect as a depth sensor for activity recognition. Accordingly, we did not mix these sensors to annotate 3D object trajectories and instead developed two separate automated annotation methods based on the manually labeled bounding box tracks:

[10] https://github.com/opencv/cvat accessed: June 15, 2022

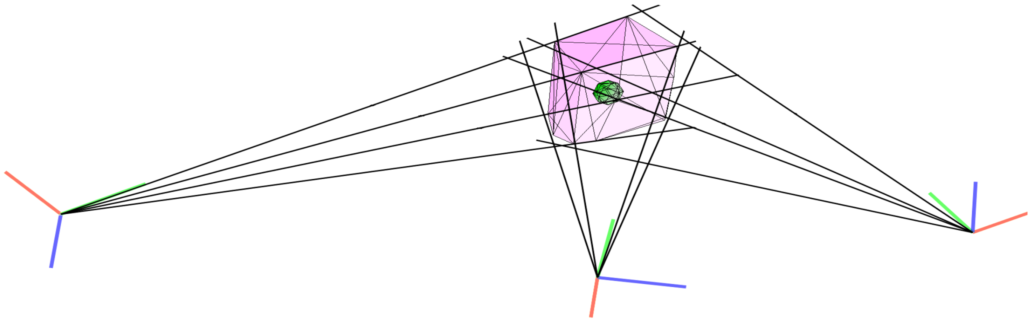


Figure 3.18: Overview of the method for bounding box triangulation via constructive solid geometry. Coordinate systems indicate camera viewpoints, black lines indicate frustum edges, the pink polygon indicates their intersection, and the green sphere indicates the resulting 3D location.

Multi-view systems usually rely on triangulation of corresponding points in each image to generate 3D data. However, this was difficult in our case because objects like the newspaper are large and their shape may vary. They were often visible from different sides in each view without common surfaces to find accurate corresponding keypoints. Instead of using keypoints we therefore took a volumetric approach and used the manually labeled object bounding boxes of the three NIR cameras to reconstruct the 3D position of the objects. Figure 3.18 shows an overview of the method. The bounding box of an object in a single view corresponds to a volume in 3D space containing the object. This volume forms a pyramid where the tip corresponds to the origin of the camera and the four edges correspond to the bounding box corners. When constructing these pyramids out of bounding boxes in multiple views they intersect in world space indicating the volume of space that is occupied by the object. To compute this intersection of the 3D volumes we used constructive solid geometry which is a method used in computer graphics and computer-aided design (CAD). It allows the construction of complex 3D volumes using boolean operators [Thi87]. Applying this method to our data resulted in a bounding volume of the object in the form of an arbitrarily shaped polygon mesh. We averaged the vertices of this mesh to determine the final 3D position of the object. We used pycsg[11] to compute the intersection of the 3D volumes.

Depending on the size and shape of the object the resulting absolute position will vary in accuracy. We were not able to evaluate the performance quantitatively based on our data. Instead, we manually checked random samples of the data comparing the resulting 3D location to the point cloud of the Kinect. The Kinect was not in any way involved in the annotation process for the triangulation-based system, it just served as a means for comparison because it provides absolute 3D coordinates of the surface of the annotated objects. In general, the annotation results reflected the position of the object well when compared to the point cloud of the Kinect.

[11] https://github.com/timknip/pycsg accessed: June 15, 2022

Depth-based annotation relied just on the bounding box of the object in the NIR image of the Kinect camera and the corresponding depth image to determine the 3D position of the object. However, the bounding box may contain parts of the background as well as other foreground elements. Determining the depth of an object by averaging the depth values of the whole bounding box may therefore lead to unexpected results. Instead, we took a heuristic approach to determine a mask of the foreground object in the bounding box. The heuristic determined the largest connected foreground area in the bounding box that was also closest to the camera. To achieve this, we clustered the depth values inside the bounding box using Mean Shift clustering with a bandwidth of 20 cm. The depth values of this cluster were then converted to 3D points using the camera intrinsics and averaged to determine the 3D position of the object.

Similar to the triangulation-based approach we reviewed the results by manually comparing the object's 3D position to the point cloud of the Kinect. The accuracy of both triangulation and depth-based methods were similar. However, the 3D trajectories of the triangulation-based annotation were more complete because the multi-view system had a wider field of view and provided robustness to occlusions.

3.5.7 Evaluation Metrics

Both driver body pose and activity recognition benchmarks needed standardized performance metrics to compare results of different methods. There were some particularities resulting from the recording in the car interior as well as the original experiment design that affected the way the results were evaluated. The metrics presented here for Drive&Act were also used to evaluate our two previous datasets.

The Drive&Act driver body pose benchmark allowed evaluating both 2D as well as 3D body pose methods. Most related 3D body pose estimation datasets assumed that the ground truth as well as the estimated results are complete without missing keypoints. In our case this was not an option because occlusion of body parts affected both the estimated results as well as the manually labeled ground truth. Some of the common performance metrics could deal with this better than others. We therefore used more than one metric to combine their strengths:

Object Keypoint Similarity (OKS) was the main metric introduced by the COCO benchmark [Lin14] for 2D human body pose evaluation. Its basis is the Euclidean distance between estimated and ground truth keypoints measured in pixels. There are additional weights to model both the annotation error and the acceptable quality for each keypoint. The main metrics are the mean average precision (AP) and the average recall (AR) over 10 OKS thresholds. The metric is robust to missing keypoints.

Mean per joint position error (MPJPE) measures the mean Euclidean distance in meters or millimeters between measured and ground truth keypoints of the 3D body pose. For multiple frames it is defined as the mean of the result for each pose. This metric is susceptible to missing keypoints because they can distort the results severely. We therefore only considered keypoints that were valid for measured and ground truth body poses to compute this metric.

Availability measures what fraction of the ground truth keypoints could be retrieved by the method. This is complementary to MPJPE which only considers the position error of retrieved keypoints while the availability metric does not consider the quality of the retrieved keypoints.

Percentage of Correct Keypoints (PCK) considers estimated keypoints correctly classified if their distance to the ground truth is within a threshold. We followed Mehta et al. [Meh17a] and determined the area under the curve (AUC) with thresholds of 1 to 150 mm (PCK_{1-15}). This metric combines aspects of MPJPE and availability because missing keypoints are treated the same as keypoints with an error larger than the largest threshold. High scores can be achieved with a combination of good MPJPE results and high availability.

Our activity recognition benchmarks differed from related activity recognition datasets because of their imbalanced class distribution. The commonly used accuracy metric would therefore favor methods that detect common activities correctly while the results of less frequent activities would not have much impact. We therefore relied on performance metrics that weigh each class the same independent of their frequency in the dataset:

Balanced Accuracy is defined as the average of the top-1 recognition rate for every category. For balanced datasets this metric is the same as the commonly used accuracy metric. This was the benchmark metric for the Drive&Act dataset. To determine the overall performance, we aggregated the results of each split and calculated the metric globally. This is statistically more stable compared to averaging the metric of each split if there are few samples for some classes making the distribution between splits imbalanced as well.

F1-Score is the harmonic mean between precision and recall. The metric is also robust for imbalanced data. It was used initially for our evaluation on the InCarIn dataset. For this thesis we only reported these results as reference and evaluated all methods using balanced accuracy.

3.5.8 Summary

The Drive&Act dataset is a large-scale public dataset for activity recognition in automated cars. Compared to related datasets, as of writing this thesis, it includes the most labels related to driver activities in three hierarchy levels. It was also one of the first large scale automotive datasets for activity recognition and is still the only dataset focused on automated driving with activities that could not be performed safely in manually controlled cars.

The dataset consists of 29 video streams recorded of 15 people including 9.6 million frames. The recording setup consisted of six cameras in a calibrated multi-view system resulting in eight camera streams covering three image modalities (Color, NIR, Depth). Apart from the video data there are additional annotations that can be used as input for driver activity recognition methods including a 3D model of the interior with 28 zones, 9 annotated objects with bounding box tracks as well as 3D trajectories and automatically labeled 3D driver body pose data.

The activity recognition benchmark includes three annotation levels focusing on 12 long term tasks, 34 fine-grained activities and driver-object interactions (Atomic Action Units) with 5 activity classes, 17 objects and 14 location labels. Since we specifically aim to rate generalization to new drivers, we evaluate exclusively on people previously unseen by the classifier. We randomly divide our dataset into three splits based on the identity of the person behind the steering wheel. For each split, we use the data of ten subjects for training, of two subjects for validation, and of three drivers for testing (i.e., 20, 4 and 6 driving sessions, respectively). Since the annotated actions vary in their duration, we divide each action segment in chunks of three seconds or less and use them as samples in our benchmark. In addition, the camera system offers the opportunity for a cross-view and cross-modal evaluation.

3D driver body pose estimation is a prerequisite for our intended activity recognition methods. While Drive&Act provides 3D driver body pose labels for the purpose of activity recognition, their accuracy as well as the accuracy of the driver body pose methods proposed in the next chapter could not be evaluated without high quality ground truth labels for this task. According to our review of related work there were no suitable public datasets. We therefore extended Drive&Act with a 3D driver body pose benchmark consisting of 1500 manually labeled unique 3D driver body poses from different activities of the dataset. In addition, the benchmark also provides 6000 manually annotated 2D driver body poses from four views that were triangulated to generate the 3D ground truth.

Overall, Drive&Act is a dataset providing multiple benchmarks on a wide variety of data with room to explore additional research directions in the future.

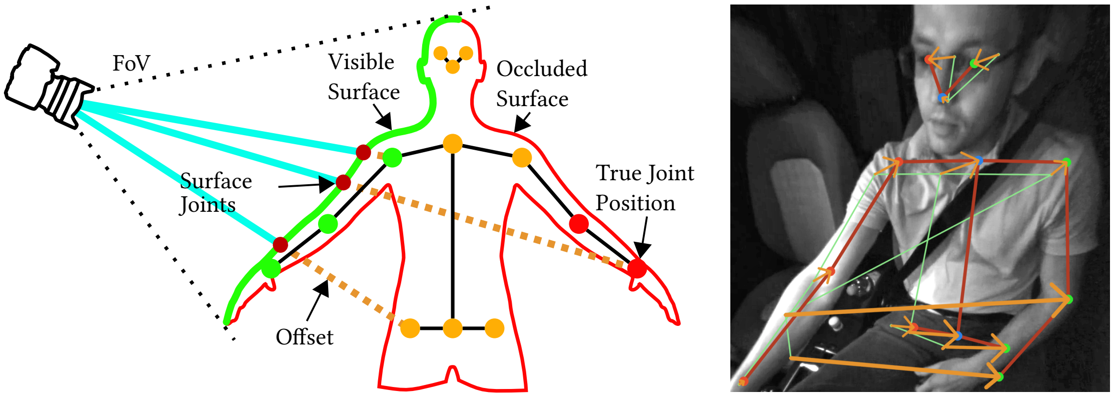
4 3D Driver Body Pose Estimation

The 3D driver body pose is the primary input feature of our activity recognition methods. The overall goal for all our approaches to determine the 3D driver body pose was real-time capability to be able to run the methods in test vehicles with limited hardware resources. Accordingly, our first approach did not rely on deep learning but was instead based on random decision forests (see Section 4.2). The second approach made use of the advancements in deep learning-based 2D body pose estimation as well as the increased computing capabilities making it possible to run these methods in real-time (see Section 4.3). Both methods dealt with the scarcity of training data in the automotive context, discussed in our review of related work, as well as with the challenges for 3D body pose estimation using depth images in different ways.

4.1 Challenges for 3D Body Pose Estimation Using Depth Cameras

Depth cameras, especially the first version of the Kinect [Sho11], helped to introduce high quality 3D body pose estimation for consumer applications. However, there were also downsides to using depth cameras for the task compared to multi-view camera systems.

The depth image is only a 2.5-dimensional representation which means the data contained in the depth image only describes the scene from a single point of view. Therefore, there is no depth data for body parts that are occluded by the environment or by other parts of the body (self-occlusion). In unconstrained environments, where people can turn around, there are therefore angles where half of the body occludes the other half (see Figure 4.1a). This causes challenges for depth image-based 3D human body pose estimation. In addition, keypoints of the human body are inside of the body while the depth image depicts the surface. So, all keypoints are fundamentally occluded by the nearest surface of the body. Therefore, there is a depth offset between the observable projection of a keypoint on the surface and its true position. For some keypoints this offset is small and mostly fixed (e.g., wrists), for other keypoints it depends heavily on the body proportions of the person as well as the viewpoint of the camera (e.g., hips, shoulders). Self-occlusion exacerbates this problem because the depth value from the depth image associated with a keypoint might be from a different body part instead of from the closest surface. Figure 4.1a illustrates



(a) 2D description. **(b)** Offsets determined by PoseFix.

Figure 4.1: (a) Two dimensional depiction of the challenges for depth image-based driver body pose estimation. (b) Handling of such a case by our method.

this case. For body keypoints on the side of the camera (green) the closest surface is observable making the offsets easier to predict. The torso and head (orange) are harder to locate, if occluded, because the observable depth value can change drastically based on the movement of the occluding arm. However, in the case of 3D driver body pose estimation there are strong priors for these keypoints because of the fixed seating position. If one arm occludes the other (red) it is often hard to predict the true 3D position of the occluded keypoints because there are different possible, and equally likely, positions. The depth cameras of the InCarIn activity dataset as well as Drive&Act dataset are located at the A-pillar on the co-driver side. The resulting side view therefore increases the difficulty of determining the location of keypoints on the left side of the driver's body. Figure 4.1b shows an example where our proposed PoseFix method successfully handles this case.

Finding an accurate solution with a single depth camera is not possible for severe occlusions because too much information is lost. Multi-view triangulation-based methods do suffer from these challenges to a lesser degree because they observe the scene from different viewpoints making them more robust to occlusions and making it easier to estimate the true position of body keypoints inside the body.

While our first method for 3D driver body pose estimation based on random decision forests does not handle these challenges explicitly (see Section 4.2) our next method, called DepthFix, can handle both cases in many situations (see Section 4.3). In addition to our depth image-based contributions, we could also show the advantages of multi-view camera-based methods (see Section 4.3.2) which were therefore used to annotate the Drive&Act dataset with 3D driver body pose ground truth data (see Section 3.5.4).

4.2 Random Decision Forests for 3D Driver Body Pose Estimation

The following method for 3D driver body pose estimation was developed in the context of the InCarIn project (2014-2017). The goal was to track all people inside the test vehicle of the project, which was also used to record the InCarIn activity dataset (see Section 3.4). This required the processing of three Kinect cameras mounted on the A-pillars to capture the front seats as well as on the ceiling to monitor the rear seats. However, in the time frame of the project real-time performance was still challenging for deep learning-based methods for human body pose estimation. In addition, the computing resources inside the test vehicle were limited. Considering the project constraints and according to our research of the state of the art, the most promising approach was therefore 3D driver body pose estimation using random decision forests. In addition, Shotton et al. [Sho11] showed that simulated depth data could be sufficient to train our approach which also alleviated the lack of suitable public training data as we show in the following section.

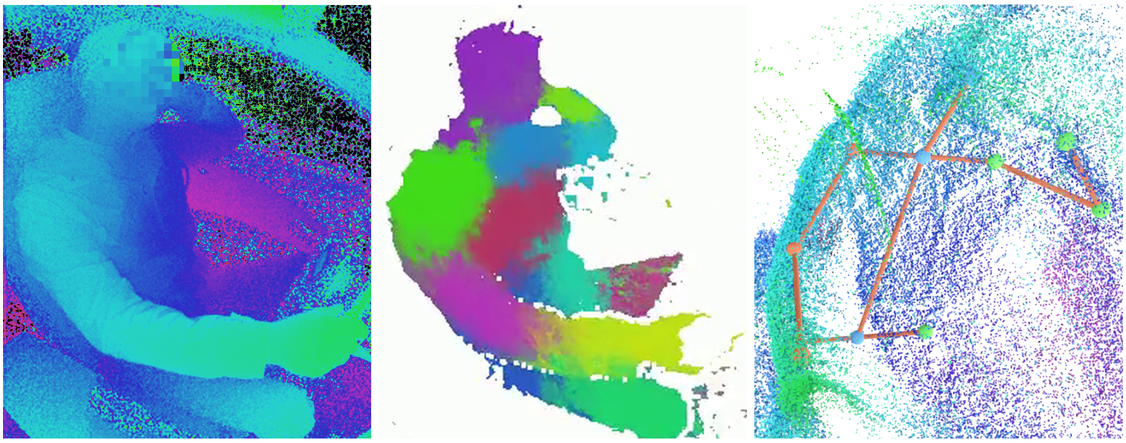
The resulting method was published at the International Conference on Intelligent Transportation Systems (ITSC) in 2017[Mar17a] © IEEE, 2017. The following chapter is based on this publication.

4.2.1 Method

Our method depends only on depth data to estimate the 3D body pose of the driver (see Figure 4.2a). It uses random decision forests to estimate dense body part labels. Our system processes the data in multiple steps that we describe in detail in the following section. At first the depth image is pre-processed to remove noise, then the driver is separated from the background using a 3D background model of the car's interior. Afterwards each point of the segmentation is classified with a random decision forest to determine body part labels (see Figure 4.2b). Finally, these pixel-wise labels are clustered to determine the body pose of the driver (see Figure 4.2c).

Preprocessing

Depending on the depth sensor and its integrated preprocessing, depth images can exhibit different noise characteristics. To clean up the image, we first use thresholding on the near-infrared image corresponding to the depth image to remove underexposed areas which indicate depth values with increased noise. Afterwards the system applies a median filter and an edge-aware Gaussian smoothing filter with size s_{filter}. The filtered depth image is then transformed to a point cloud using the intrinsic parameters of the camera.



(a) Input depth image **(b)** Segmentation and body part labels **(c)** 3D driver body pose

Figure 4.2: The stages of the body part detector showing the input depth image (a). (b) The segmentation with body part labels and (c) the resulting 3D driver body pose overlayed on the point cloud.

Segmentation

After preprocessing the image, the system separates the image points belonging to the driver from the car's interior. To achieve this, it generates a voxel-grid representation of the empty interior to efficiently segment the driver at test time.

To create the background model multiple frames are accumulated in a voxel-grid with resolution r_v. Once per frame and voxel a counter is incremented if any point of the input is within the voxel. The final model is then defined as follows:

$$M_i = \begin{cases} \text{interior} & \text{if } \frac{A_i}{n_{\text{seg}}} > t_{\text{seg}} \\ \text{free space} & \text{otherwise} \end{cases} \tag{4.1}$$

where M is the background model, A is the accumulator voxel-grid, i is the index of a voxel, n_{seg} is the number of accumulated frames and t_{seg} is a threshold between zero and one controlling the noise susceptibility of the background model. Using a high threshold is preferable because it lowers the impact of sensor noise on the background model. Setting the threshold too high will cause artifacts because depending on the material of the interior some surfaces only produce noisy but consistent depth data that should still be captured in the background model. Setting the threshold too low causes voxels to be marked as background because of noise in the depth image.

At test time the algorithm determines the enclosing voxel for each point of the point cloud. If the voxel is not marked as belonging to the background then the point belongs to the segmentation of the driver (see Figure 4.2b).

Body Part Labeling

We use random decision forest to determine dense body part labels for all foreground points of the depth image similar to Shotton et al. [Sho11]. Random decision forests are effective multi-class classifiers consisting of multiple random decision trees. Each tree is trained and evaluated independently. Each inner node of the tree contains a learned weak binary classifier determining how each sample is propagated to the child nodes. Our classifier is based on depth comparisons:

$$S_l = \{x | x \in S \wedge d(p_{uv} + \frac{o_1}{d(p_{uv})}) - d(p_{uv} + \frac{o_2}{d(p_{uv})}) < t_{\text{label}}\} \tag{4.2}$$

$$S_r = S/S_l \tag{4.3}$$

where $\{S\}$ is the set of samples split by a node into samples for the left $\{S_l\}$ and right $\{S_r\}$ child node. Each sample consists of an image position p_{uv} and a ground truth body part label when training the tree. $d(p_{uv})$ is the depth at position p_{uv}, o_1 and o_2 are learned offsets in image coordinates and t_{label} is a learned threshold. $d(p_{uv})$ is set to a large penalty value d_{outside} if the evaluated position is outside of the segmentation.

When training a random decision forest for body part labeling the samples reaching each leaf of each tree t are used to determine a probability distribution $P_t(l|p_{uv})$ for body part labels l. The result of the random decision forest is determined by averaging the results of all trees T:

$$P(l|p_{uv}) = \frac{1}{|T|} \sum_{t \in T} P_t(l|p_{uv}) \tag{4.4}$$

Figure 4.2b shows the result of the body part labeling step.

Pose Estimation

To determine the 3D body pose of the driver the pixel-wise labeled points must be combined. Averaging all points of the segmentation weighted by their probability for a body part would result in severely degraded results because of misclassified points. Instead, we rely on clustering of the body part labels using the most probable body part for each point based on the maximum of each point's probability distribution. We then apply connected component clustering using both the label of each point and the Euclidean distance between neighboring points to create clusters. Only points with the same label that are close to each other are assigned to the same cluster:

$$C_i = \{p, q | l(p) = l(q) \wedge ||p - q|| < t_{\text{cluster}}\} \tag{4.5}$$

where C_i is the i-th cluster p and q are neighboring points in the depth image, $l(p)$ is the label of p and t_{cluster} is the threshold determining the maximum distance of neighboring points belonging to the same cluster.

For each resulting cluster i we determine both the centroid J_i and a cluster body part weight W_i:

$$J_i = \frac{1}{W_i} \sum_{p \in C_i} w(p) p \tag{4.6}$$

$$W_i = \sum_{p \in C_i} w(p) \tag{4.7}$$

where $w(p)$ is the probability that point p belongs to the body part p.

Each cluster centroid and corresponding cluster weight is a hypothesis of a body part's location. It is likely that there are multiple hypotheses for each body part. We choose the centroid with the highest cluster weight as the position of each body part. Our method does not produce body part labels for the neck and pelvis keypoints instead we determine their position by averaging the shoulder and hip keypoints respectively.

4.2.2 Random Decision Forest Training

Training random decision forests for body part labeling requires a large number of depth images with body part ground truth labels. Labeling this data manually is more time consuming than just labeling keypoints in the image because it requires pixel-level annotation of regions. We therefore explored different methods to create data for training with automated annotation. We both collected real sensor data using low fidelity motion capturing for labeling as well as synthetic data. In the following section we first describe the training method for random decision trees followed by our methods for training data collection.

Random decision tree training method

The goal of training a random decision tree for body part labeling is choosing a weak classifier for each node of the tree to split the training samples into two parts with the cleanest possible distribution of labels. The weak classifier is defined in Equation 4.2. Its parameters are the two offset vectors o_1 and o_2 as well as the threshold parameter t_{label}.

Instead of a gradient descent approach to find these parameters random decision trees rely on randomly choosing these parameters within user-defined bounds. The training method generates a user-defined number of random weak classifiers for each node and chooses the classifier that performs best from this set. The parameter bounds influence how well the detector can perform, and the number of random classifiers tested for each node influences its ability to generalize to new data. While a high number of classifiers can improve accuracy it can also increases the risk of over-fitting on the training data.

The whole random decision tree can be trained using a greedy approach starting at the root and proceeding to each child node repeating the following steps:

1 Generate a set of n weak classifiers randomly choosing features o_1, o_2 and t_{label}.

2 Compute the split of the training samples S into S_l and S_r for each weak classifier

3 Determine the information gain (IG) for each weak classifier:

$$IG = H(S) - \sum_{c \in \{l,r\}} \frac{|S_c|}{|S|} H(S_c) \tag{4.8}$$

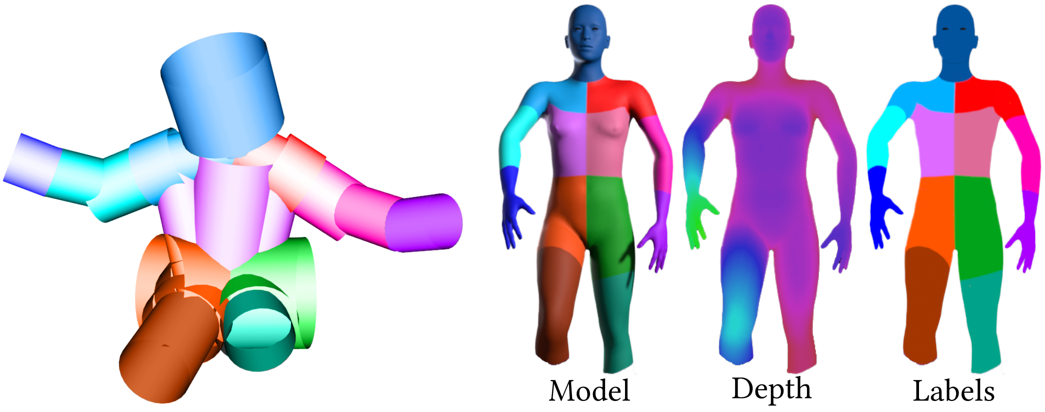
where $H(S)$ is the Shannon entropy of the histograms of body part labels.

4 Choose the weak classifier with the largest information gain and continue with S_l and S_r if the depth of the tree has not reached the maximum

We train each decision tree of the random decision forest separately. The randomness introduced in each tree increases the chance that each tree focuses on distinctive features.

Real training data

Our first goal was to test the performance of the method on the Driver Depth Pose dataset (see Section 3.3). We collected real world training data for this experiment. This involved collecting data from the viewpoint of the interior mirror using the Melexis Sensor because this was the camera viewpoint of the benchmark dataset. To avoid manual labeling, we used a Microsoft Kinect 2 for Windows and its 3D body pose estimation method as a low fidelity motion capture device. Both cameras were calibrated with markers such that they have a common frame of reference. With this setup it was possible to capture the depth image from the Melexis camera and the 3D human body pose using the Kinect as labeling system. However, the goal was to produce body part labels using an automated method. To achieve this, we constructed a volumetric body model based on cylinders (see Figure 4.3a). For each point of the point cloud of the Melexis sensor we determined the intersection with this volumetric model. Depending on which cylinder the point intersected it was assigned the respective ground truth body part label. If a point was outside



(a) Cylinder model

(b) Synthetic data sample

Figure 4.3: (a) The cylinder model used to annotate real data. (b) Synthetic data sample depicting the model with labeling texture, resulting depth image (false color) and body part label annotations.

of the volumetric model the point was discarded. With this method we created both the body part labels and the segmentation of the training images.

The training data was collected outside of the car to be able to position the Kinect sensor, so its 3D body pose estimation method worked well. This was not possible in the car's interior. This approach was only viable because our method relied on a segmentation of the driver for body part labeling. Accordingly, the surroundings did not matter for collecting training data.

The accuracy of this annotation system was limited because of the coarseness of the cylinder-based model. We therefore only labeled 12 large regions (shoulders, elbows, hands, hips, knees, torso, and head). We collected 16 000 training images from 5 different people (all male) of different body type and height. Two people were collected with and without thick winter jackets to change their shape in order to increase the variance of body proportions in the training set. All participants performed the different poses from the Driver Depth Pose dataset multiple times with variations in their execution. In addition, participants performed random poses while seated to improve the coverage of the degrees of freedom of the upper body.

Synthetic training data

While the method to collect real world data worked well for the Driver Depth Pose dataset, it was an involved process that would have to be repeated for other sensor configurations. However, our final goal for 3D driver body pose estimation was activity recognition using the InCarIn Activity dataset as well as 3D body pose estimation of all passengers in the test vehicle of the InCarIn project. This would have required additional data collection for the Microsoft Kinect 2 for Windows from multiple camera viewpoints. Instead, we decided to use synthetic data for the task. This both enabled us to create more varied data with less effort as well as to create ground truth body part labels with higher accuracy.

Generating synthetic data required a rendering system, different models of the human body with varying proportions, as well as motion capture data to move the models. We used Blender as a rendering engine to generate depth images as well as body part labels. Varied human body models were generated with the parametric body model of Make-Human (see Figure 4.3b). There was no public motion capture data from people inside vehicles and only scarce data for sitting people in general. Like our method for real world data creation, we therefore used the Kinect and its 3D body pose estimation method to collect motion capture data. However, in this case both the camera images as well as the recording environment were not important. We therefore only collected motion data from one person because body type and height were determined by the rendering system and were independent from the motion capture data. Overall, we collected 80 000 3D body poses with this setup. This included the movements from the Driver Depth Pose dataset, the activities from the InCarIn activity dataset as well as random movements of the upper body. However, this data was recorded with a high frame rate so the body pose did not change much from frame to frame. We therefore selected a subset of 11 000 unique poses from this data with a distance of at least 10 cm in one keypoint to all other selected poses. This was the same process used to select frames for the Drive&Act driver body pose benchmark (see Section 3.5.4).

For each rendered training image, the data creation pipeline chose a random 3D body pose from the collected motion capture data, one of 16 human models (eight male, eight female) as well as a random transformation augmenting the rotation, position, and height of the person. The virtual sensor was configured with the camera parameters of the Kinect 2 for Windows and its position resembled the depth camera position of the InCarIn Activity dataset as well as Drive&Act dataset.

Overall, the final synthetic training dataset consisted of 100 000 samples including depth images as well as body part label annotations with 13 regions.

Table 4.1: Parametrization of the random decision forest-based 3D driver body pose estimation method for different datasets.

Algorithm Stage	Parameter	Depth pose dataset	Drive&Act and InCarIn
Proprocessing	Noise filter size: s_{filter}	3	
Segmentation	Voxel grid resolution: r_v	5 cm^3	3 cm^3
	Background threshold: t_{seg}	0.3	0.8
	# frames for background: n_{seg}	100	
Part labeling training data	# Images	18 000	100 000
	# Labeled regions	12	13
	Type	Real	Synthetic
Part labeling training	# Trees	5	3
	Tree depth	20	
	# Offset pairs: o_1, o_2	1500	
	Offset Range: o_1, o_2	−100 to 100 px	−120 to 120 px
	# Thresholds: t_{label}	50	
	Threshold range: t_{label}	−1 to 1 m	
	Penalty value: d_{outside}	100 m	
Body part label clustering	Max neighbor distance: t_{cluster}	0.1 m	

4.2.3 Implementation Details

The approach has some parameters that need to be chosen based on the target camera as well as training dataset. For most of the development of this algorithm labeled data was scarce or not available. Most of the parameter search was therefore done by inspecting results of the different stages of the method on unlabeled data. In addition, for some parameters we used statistics of the training dataset to choose suitable values.

We use two configurations of the method in this thesis. One configuration is trained on real data and is tested on the Driver Depth Pose dataset. The second configuration is trained on synthetic data and is tested on the more complex Drive&Act driver body pose benchmark. In addition, this model is used to generate the 3D driver body poses used for evaluating on the InCarIn Activity dataset. Table 4.1 lists all parameters of the approach and their values for both configurations.

Preprocessing and segmentation depend on the sensor and its noise characteristics. The Melexis sensor of the Driver Depth Pose dataset produces depth data with a high noise level. Accordingly, the voxel grid resolution (r_v) and the threshold to classify a voxel (t_{seg}) as being part of the background is lower compared to the configuration for the Kinect Sensor which produces depth images of higher quality. In both cases we use $n_{\text{seg}} = 100$ frames to create the background model of the empty car.

The parameters for body part training depend on the viewpoint of the camera and its resolution as well as the number of body part labels and the size of the dataset. We choose the parameters based on the findings of Shotton et al. [Sho11] as well as statistics of our training datasets. The number of trees does not influence the body pose estimation result much in our experiments. We therefore use five decision trees for the Driver Depth Pose dataset but only three trees for the other configuration to reduce the processing time for real-time use. We set the depth of the random decision forest to 20. This has a large impact on the performance as we will show in the next section. Deeper trees perform better in general. However, there are limits because each node splits the number of training samples into two parts so the deeper the decision tree the more training samples are needed to still be able to generate meaningful statistics in leaf nodes. The main parameters regard the creation of the random weak classifiers for the training of each random decision tree node. This requires choosing a suitable pair of offsets (o_1, o_2) and a corresponding threshold (t_{label}) for the training objective. We choose 1500 random offset pairs and for each pair we test 50 random thresholds. If too few weak classifiers are sampled the output of the decision tree is noisier. Large values on the other hand mostly increase training time without negatively affecting the results. The number of necessary samples also depends on the range of values the offsets and thresholds are sampled from. For a larger sampling range more samples are necessary to find suitable candidates. We choose the range for the offsets based on the average torso length in pixels for each dataset. This allows the decision tree to sample a large neighborhood while decreasing the chance to generate offset pairs that are outside of the segmentation of the driver.

We trained the random decision forest with a custom distributed training system on 100 cores. It took 24 h to train the configuration based on synthetic data. We also performed a runtime test on the hardware used in the test vehicle of the InCarIn project (Intel Core i7-4700MQ 2.4 Ghz, Nvidia Geforce GT 730M with 1 GB video memory). The real-time system was implemented in C++. The only part that was optimized for parallel execution on the CPU or on the graphics card using CUDA was the inference from the random decision forest. Everything else was executed single-threaded on the CPU. We reached 23 fps using a single thread, 37 fps with multiple threads and 51 fps using the graphics card. We could therefore achieve our design goals regarding real-time performance stated at the start of the chapter.

4.2.4 Evaluation on the Driver Depth Pose Dataset

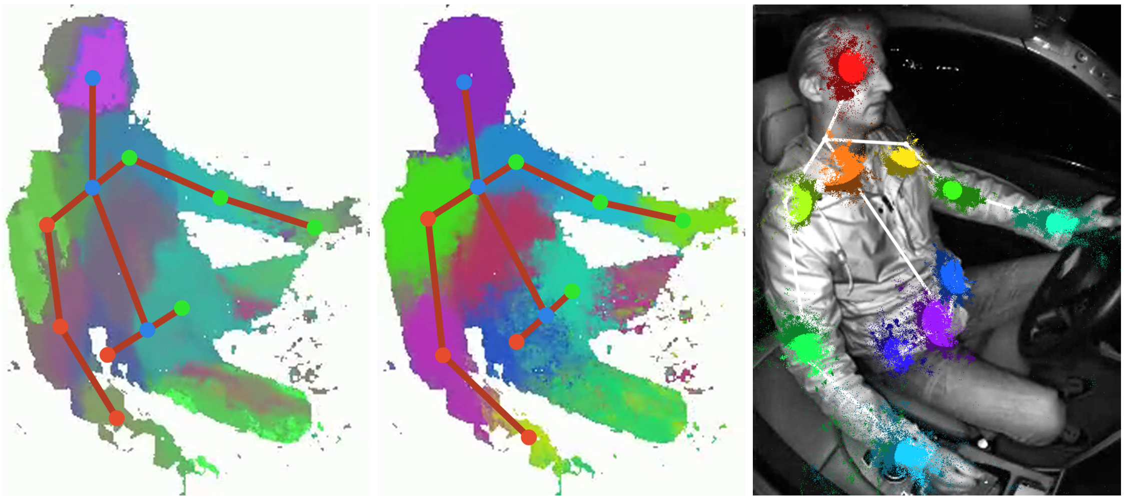
The primary dataset for evaluating the approach outlined in this thesis is the Driver Depth Pose dataset (see Section 3.3). We train the method on real data collected from the same sensor (see Section 4.2.2) using the parametrization presented in the last section. The performance metrics that were used are presented in Section 3.5.7.

Table 4.2: Results for 3D keypoint detection on the Depth Pose dataset using random decision forests (DF) with T trees and depth D. Best results are marked bold. The columns on the left show statistics per keypoint while the columns on the right show statistics per body pose.

Config.	head	lShoulder	rShoulder	neck	lElbow	rElbow	lWrist	rWrist	lHip	rHip	midHip	mpjpe	PCK_{1-15}	avail.
	mpjpe [mm]											[mm]	[%]	[%]
DF T5 D5	**26**	69	55	**76**	46	122	142	189	78	83	112	92.3	40.5	100
DF T5 D10	29	**62**	57	80	35	95	53	76	69	64	101	65.3	51.3	100
DF T5 D15	31	64	49	84	**31**	72	48	62	64	59	96	59.8	54.4	100
DF T5 D19	31	63	**48**	86	**31**	69	47	54	61	57	**94**	58.1	55.0	100
DF T5 D20	31	63	**48**	86	32	**68**	**46**	**53**	**60**	**56**	**94**	**57.9**	**55.2**	100

We evaluate the performance of the resulting model for different random decision forest depths. To achieve this, we train the forest to a maximum depth of 20 but only evaluate it partially at test time. Table 4.2 shows the results. The random decision forest with the full depth of 20 achieves the best result with a mean per joint position error of just 57.9 mm and a PCK_{1-15} score of 55.2 %. The method is in general able to detect all annotated keypoints (100 % availability) of the dataset. Overall, the mean per joint position error (MPJPE) decreases by 37 % with increasing depth of the random decision forest. However, while the initial performance gains are large the differences get smaller with increasing depth. Keypoints that move a lot (i.e., wrist) or are often partially occluded (i.e., hips) are harder to detect and therefore benefit the most from an increase in depth. On the other hand, keypoints that are mostly static, like the head and shoulders, can already be detected with high accuracy at a low depth of the decision forest. This indicates that they are easier to detect requiring fewer weak classifiers to arrive at the correct body part label. We would expect that the left side of the body performs worse because it is further away and potentially occluded by the right arm. This is not the case. However, it is not possible to show this effect with the Driver Depth Pose dataset because in case of occlusion it was also not possible to create ground truth labels. This is also the reason the left elbow performs so well because it could only be annotated, and therefore evaluated, in a small region if it was not occluded at the side of the torso. We can show that the left side of the body indeed performs worse on the Drive&Act driver body pose benchmark because it was labeled using a multi-view camera system including labels for occluded keypoints (see Section 4.4.2).

Figure 4.4a and Figure 4.4b visualize the effect of reducing the depth of the random decision forest. To create these images, we assign a color to each label and produce the weighted sum of the colors based on the label probability distribution for each pixel. Muted or gray colors indicate an even probability score, while bright colors indicate a clear decision. Reducing the depth decreases the overall confidence of the random decision forest, indicated by the muted colors. Some regions, like the head, that also performed well at low random forests depth, are still clearly visible. In many cases the reduced confidence still suffices to determine the keypoint positions via our clustering approach.

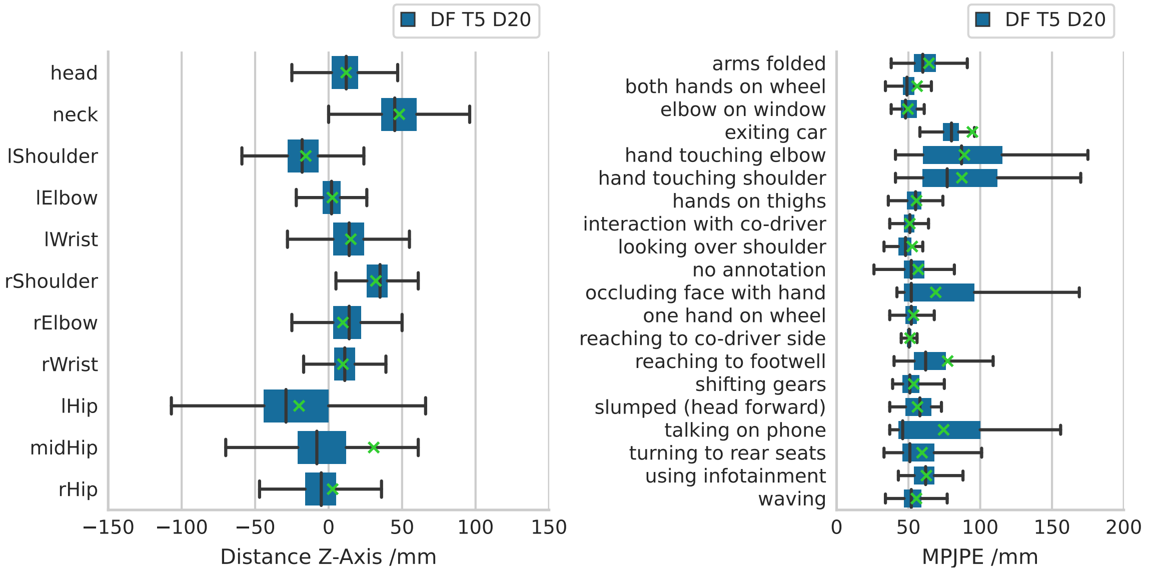


(a) Random forest depth 5 **(b)** Random forest depth 20 **(c)** x/y-error distribution

Figure 4.4: Heatmap of the random decision forest with depth five (a) and depth 20 (b). (c) Keypoint error distribution of the x-axis and y-axis reprojected onto the image. Bright colors represent 50 % of the samples with the smallest Euclidean distance to the ground truth. The reference ground truth body pose is depicted in white.

Figure 4.4c depicts the error distribution of the approach for the x-axis and y-axis. It shows that the estimated shoulder keypoints and neck keypoint are offset further down and to the body center compared to the ground truth. This is expected because our method determines the centroid of the shoulder cluster as keypoint position. The further these clusters reach down the torso the more the keypoints therefore move in this direction as well. The neck keypoint is determined by averaging the shoulder keypoints and is therefore affected by the same problem. In addition, the ground truth positions the neck further back at the base of the skull which also causes a systematic deviation of the z-axis (see Figure 4.5a).

The error distribution on the z-axis, which corresponds to the values in the depth image, should show the challenges depicted at the start of the chapter regarding surface keypoint positions and occlusions. Figure 4.5a depicts the results. The hips show the largest spread of errors on this axis. This is consistent with our general results and highlights their detection difficulty caused by occlusion by the right arm. Overall, the errors are spread around zero. We would expect a systematic negative offset because our method does not consider the difference from the surface of the body, depicted by the depth image, to the true keypoint positions. However, with the recording setup of the Driver Depth Pose dataset we were only able to label surface keypoints of visible body parts because the offset could not be determined based on the single recorded depth sensor. An evaluation on this data cannot show this effect, but we can show it on the Drive&Act driver body pose benchmark (see Section 4.4.2). The evaluation on the Driver Depth Pose dataset therefore only shows the performance of the method for visible keypoints.



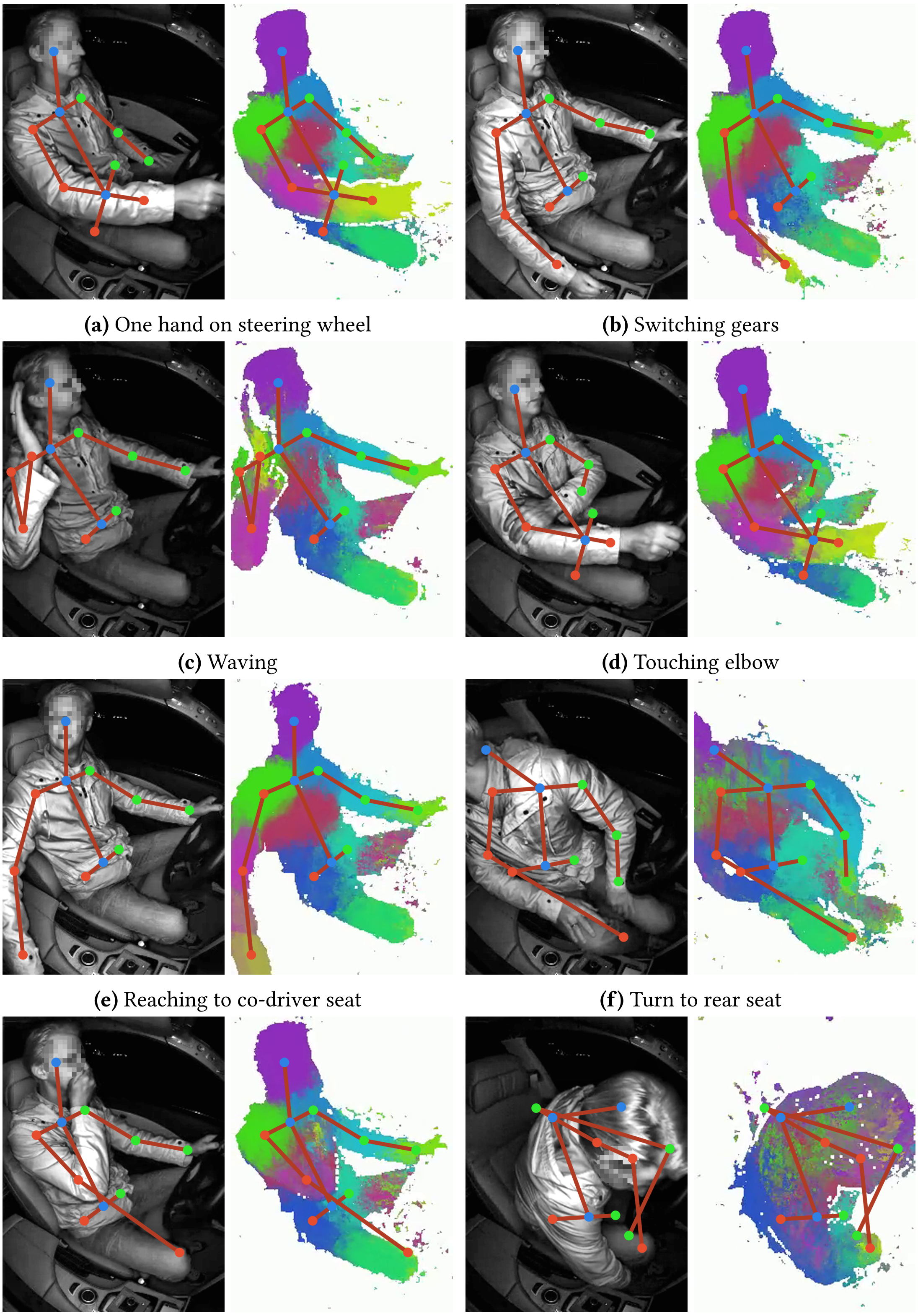
(a) Error z-axis

(b) Error activities

Figure 4.5: Statistics of the random decision forest-based body pose detector on the Driver Depth Pose dataset.

Figure 4.5b show the overall result of the 3D body pose estimation for each action. The median MPJPE of most actions is around 50 mm. Notably the actions *hand touching elbow* and *hand touching shoulder* perform worst. These are difficult to estimate because depending on the test participant they were executed with the arms close to the body. Because of the noise level of the sensor, it is then hard to separate points belonging to the arms from points belonging to the torso. In our experience, these poses are challenging for 3D human body pose methods in general. Overall, the actions that perform worst also cause large occlusions of the upper body which cannot be handled well by our approach.

Figure 4.6 shows some qualitative results of the system for different actions. In most situations the system works well. A typical failure case occurs when reaching with the right hand near the head. In this case the wrist is sometimes wrongly detected at the knee (see Figure 4.6g). This does not happen every time and it also seems to depend on the body proportions of the person. However, this is the main reason for the larger errors of all actions with similar arm movements. *Reaching into the foot-well* (see Figure 4.6h) causes occlusion of most body keypoints which makes accurate detection with our approach challenging. In addition, this pose is not represented well in the training data as indicated by the noise in the visualization of the body part labels.



(a) One hand on steering wheel

(b) Switching gears

(c) Waving

(d) Touching elbow

(e) Reaching to co-driver seat

(f) Turn to rear seat

(g) Hand on face (failure)

(h) Pick up object from foot-well (failure)

Figure 4.6: Sample results for different actions including failures.

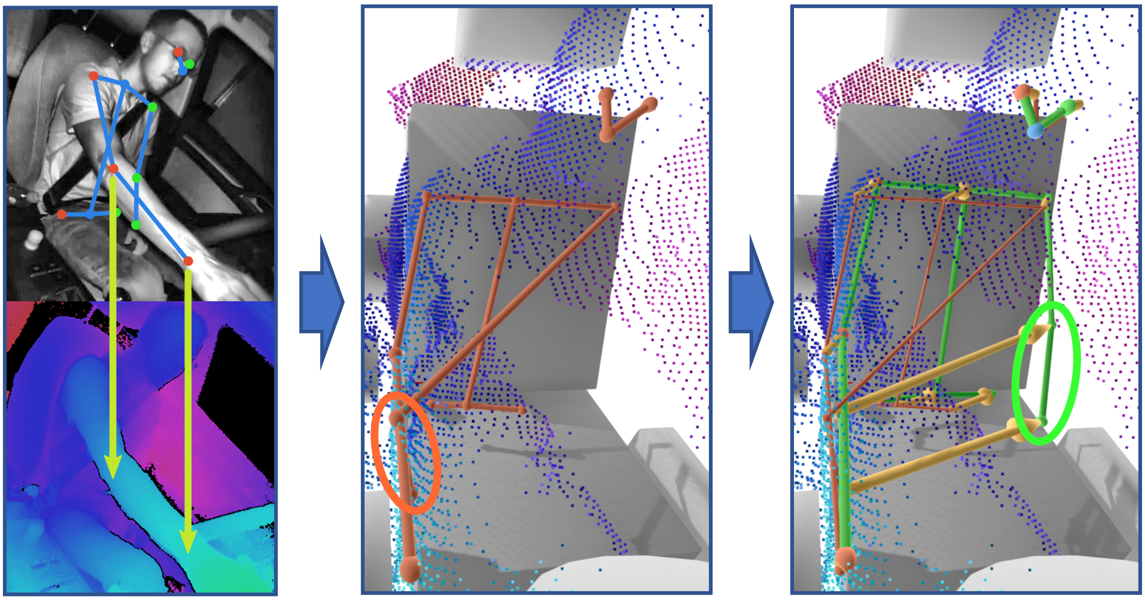
4.3 DepthFix: Driver Body Pose Estimation with Occlusion Handling

While our decision forest-based method fulfilled the needs for real-time 3D driver body pose estimation in the InCarIn project, it had some drawbacks. It relies on a background model to segment the driver in the vehicle, which is challenging in many scenarios, for example when the seats are moved by the passengers. In addition, the background model is not robust with regards to objects like jackets, laptops, food or drinks which are all part of the Drive&Act dataset. The method also disregards the peculiarities for depth image-based human body pose estimation regarding surface keypoints and occlusions (see Section 4.1). Finally, the method relies just on the depth image for 3D body pose estimation which makes it harder to detect fine details like the eyes or the nose because they stand out mostly as texture in the near-infrared image and less as shape in the depth image.

We therefore developed a new approach that accounts for these challenges using modern deep learning-based methods. However, the challenge of missing training data for 3D driver body pose estimation still remained. While our last method tackles this problem using synthetic data, our new approach separates 2D driver body pose estimation, using just near-infrared images as input, and 3D body pose estimation using depth images and 2D body poses as input. This allows us to rely on state of the art methods for 2D body pose estimation as well as large scale datasets for their training from other domains. Only the second part of the method relies on labeled data from the interior of the vehicle. Here we rely on the automated 3D driver body pose labels of the Drive&Act dataset for training. Compared to the random decision forest-based method our new approach regresses true keypoints location within the body instead of on the surface and it can handle occlusions (see Figure 4.7). In addition, it does not rely on a background model which eliminates one of the most common points of failure from our first approach.

In the following we present our proposed DepthFix method that relies on depth images for 3D driver body pose estimation and a multi-view triangulation-based approach used as baseline for our method as well as to annotate the Drive&Act dataset (see Section 3.5.4). All following 3D body pose methods rely on OpenPose for 2D driver body pose estimation. We present a brief overview of this method in the state of the art chapter (see Figure 2.1).

The following sections are based on our publication at the International Conference on Intelligent Transportation Systems (ITSC) 2021 [Mar21] © IEEE, 2021.



(a) 2D pose estimation (b) Direct pose (c) DepthFix

Figure 4.7: Overview of the DepthFix approach. (a) 2D body pose estimation on the near-infrared image (blue) followed by depth lookup (yellow). (b) The resulting direct pose result. With wrong keypoint depth estimation caused by occlusion (red). (c) DepthFix result (green) based on correcting offsets (orange) fixing the occluded keypoints (green).

4.3.1 DepthFix method

The goal of our method is to leverage the progress in deep learning-based 2D human body pose estimation for depth image-based 3D driver body pose estimation without retraining the 2D pose detector. One way to achieve this is to detect the 2D body pose of the driver on the near-infrared image of a time of flight sensor and to use the matching depth image to determine 3D coordinates for the 2D detection (see Figure 4.7a). A direct approach would just look up the depth value of each 2D body keypoint (u,v) in the depth image d and then use the inverse camera matrix C to compute the 3D keypoint K in camera coordinates:

$$K_{direct} = C^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} d(u,v) \tag{4.9}$$

However, the resulting 3D keypoints are located on the surface of the driver's body because the depth camera observes the surface of the scene. The true position of the keypoints is within the body with an additional offset to the depth image. We test the approach proposed by Shotton et al. [Sho11]. They addressed this issue by learning offset

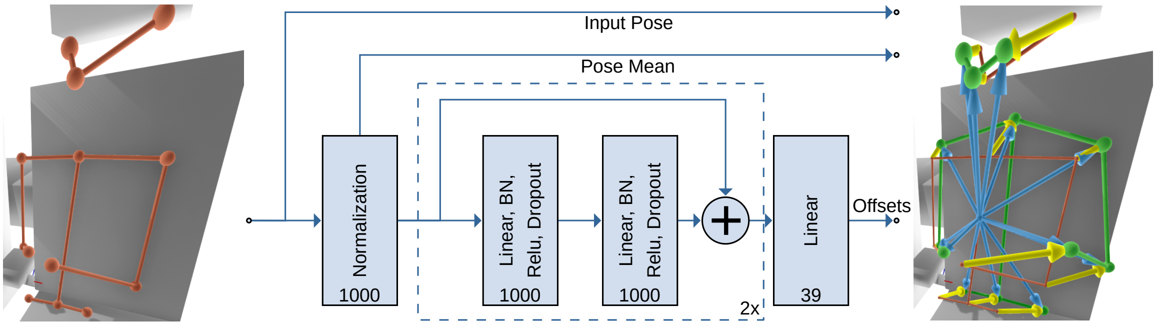


Figure 4.8: Depiction of the DepthFix method. It uses the direct 3D body pose as input(red) and produces correcting offsets via a small feed forward network resulting in an improved body pose (green). We test two different offset methods. Offsets starting from the pose center (blue) and offsets starting from each keypoint of the input pose (yellow).

vectors o_k for each keypoint on a holdout data set T with labeled keypoints K_{gt}:

$$K_{\text{offset}} = K_{\text{direct}} + o_k \quad ,\text{with } o_k = \frac{1}{N}\sum_{t\in T} K_{gt} - K_{direct} \tag{4.10}$$

This already improves results as our evaluation shows. However, there are additional challenges. Fixed offsets do not account for different body shapes or sizes. Furthermore, the depth image does not contain valid z-values for occluded keypoints. This can lead to large errors that cannot be compensated with fixed offsets (see Figure 4.7b).

Our approach therefore calculates offsets that adapt based on the 3D body pose used as input for the method (see Figure 4.7c). It can correct surface keypoint positions to true keypoint positions within the body and it tries to fix any errors introduced by occlusions by regressing correcting offsets. This is handled by a small and efficient neural network that can be applied as a post processing step after generating the 3D body pose using the direct method (see Equation 4.9). The idea is inspired by Moon et al. [Moo19] who used a neural network as a post processing step to fix common errors of 2D body pose estimators like left-right swapping of limbs. It is also inspired by Martinez et al. [Mar17b] who demonstrated that a single 2D human body pose could be lifted to 3D with a small and fast neural network as a post processing step to achieve monocular 3D body pose estimation. Our neural network architecture is similar to their approach.

Figure 4.8 shows our proposed neural network. The basic module of the network is a linear layer followed by batch normalization, rectified linear units and dropout. Two of these modules form a block of the neural network. The first block increases the feature dimensionality while all remaining blocks keep the dimensionality the same. Our neural network consists of two of these building blocks combined with skip connections. The final layer of the network is a linear layer that regresses correcting offsets.

The network uses the 3D body pose determined by the direct method as input (see Equation 4.9). The input to the neural network is therefore affected by all the problems discussed so far. The 3D keypoints are on the surface of the human body and in case of occlusions the z-value of the keypoint might be off by a large margin. We expect the network to create correcting offsets without additional information like image features or time-series data. The resulting offsets are location independent. The absolute position of the 3D input body pose is therefore not relevant and we normalize it by subtracting the mean of all valid body keypoints which is the center of the body c_p. We experiment with two different methods to generate correcting offsets:

The full regression method determines the output pose by adding the offsets to the body center c_p used for normalization (see Figure 4.8 blue). This forces the network to reconstruct the full body pose centered at the origin that is then moved to the correct location by adding the body center point c_p.

The DepthFix method applies the offsets to the corresponding input body keypoints (see Figure 4.8 yellow). This resembles the method described in Equation 4.10 but with dynamic offsets that depend on the input pose. Compared to regressing offsets from the center of the body, DepthFix can make better use of the input pose. Without occlusions the regressed offsets are small and for some body parts mostly static, so the neural network does not need to learn to reconstruct the 3D body pose from the ground up. In case of occlusions the offsets can nevertheless get larger, but they still originate from their respective input keypoint.

To train the network we use the mean squared error between estimates and ground truth. However, because both our input data as well as the labels can be incomplete, we mask the loss of missing keypoints to not penalize the network in training.

4.3.2 Baseline: Multi-View Triangulation

The triangulation-based 3D driver body pose method proposed here has multiple purposes in this thesis. It serves as a baseline on the Drive&Act 3D driver body pose benchmark where we also experiment with different multi-view setups based on the cameras of the Drive&Act dataset (see Figure 3.14a). In addition, this method was used to annotate the benchmark itself based on manually annotated 2D driver body poses and with additional reviewing and manual cleaning of the result. It was also the method used to generate the 3D driver body pose annotations published with the Drive&Act dataset for activity recognition (see Section 3.5.4). The method is not novel and relies on standard triangulation techniques. However, its application in the interior of a car as well as the following evaluation is a novel contribution to the field based on our literature review.

Our method uses the 2D driver body pose of multiple views as input and generates 3D data by triangulating each keypoint separately. In the following we therefore only present the process for a single keypoint. The goal of the triangulation is to reconstruct the 3D keypoint position X based on 2D observations x_v in at least two views $v \in V$ given a calibrated multi-camera system with known projection matrices P_v. The challenge is that the observations are not exact and therefore the lines created by re-projecting the measurements ($x_v = P_v X$) into the 3D scene do not intersect. In addition, the system is over-constrained with more than two views. We follow the linear triangulation method described by Hartley et al. [Har04] to solve this problem. They show how to construct an over-constrained linear equation system of the form $AX = 0$ where each view contributes two rows to matrix A by removing their homogeneous scale factor with a cross product $x \times (PX) = 0$:

$$x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) = 0$$

$$y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) = 0$$

$$x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) = 0$$

where $\mathbf{p}^{i\top}$ are the rows of P. Only the first two equations are linearly independent resulting in:

$$A = \begin{bmatrix} x_1\mathbf{p}_1^{3\top} - \mathbf{p}_1^{1\top} \\ y_1\mathbf{p}_1^{3\top} - \mathbf{p}_1^{2\top} \\ \vdots \\ x_v\mathbf{p}_v^{3\top} - \mathbf{p}_v^{1\top} \\ y_v\mathbf{p}_v^{3\top} - \mathbf{p}_v^{2\top} \end{bmatrix}$$

The equation system can be solved using Singular Value Decomposition. X is the unit singular vector corresponding to the smallest singular value. However, while this method produces satisfactory results in most cases it does not optimize the geometric distance between the measured points x_v and the re-projection $\hat{x}_v$ of X. We therefore use the result as a starting point for further nonlinear optimization[Gao12] of X using the sum of re-projection errors as minimization goal:

$$E(X) = \sum_{v \in V} |x_v - \hat{x}_v|_2^2$$

Finally, we discard any triangulation result if its mean re-projection error is above a threshold.

We repeat this process independently for each keypoint of the body with valid measurements in two or more views.

4.3.3 Implementation Details

Both depth and triangulation-based 3D driver body pose methods rely on OpenPose [Cao18] for 2D driver body pose estimation. The three parts of our approach are parametrized as follows:

OpenPose is used with its default model providing 25 keypoints. However, we only use 13 keypoints of the upper body depicted in Figure 4.8. OpenPose is trained on color data and requires three channels as input. We therefore generate three channel images by replicating the grayscale data. Images of camera IR 1, IR 3 and the Kinect (all mounted at the A-pillars) must be rotated by 90° so the driver is upright in the image. Without this transformation OpenPose will not work. The images are unevenly lit. We therefore test different methods to adapt the brightness. The best results are achieved by using adaptive histogram equalization (CLAHE) of OpenCV [Bra00] with a limit of 2.

The triangulation-based approach relies on OpenPose results of different combinations of at least two camera views of the Drive&Act dataset. It does not rely on any parameters except for the filter threshold to discard keypoints with high mean reprojection error. We choose a threshold value of 20 to limit filtering to large outliers.

DepthFix and its modifications require training. Because of our limited manually annotated data we use the best results of the triangulation method as ground truth for the Drive&Act dataset and use all sequences not part of the Drive&Act driver body pose benchmark for training. This approach results in a cross-person evaluation of these methods. The neural networks are implemented in PyTorch 1.4.0. We use the Adam optimizer with default parameters and train for 80 epochs using a batch size of 128 and optimizing the mean squared error. The learning rate is multiplied by 0.1 after 30 and 60 epochs. Participants 2 and 3 are used for validation. We test on the Drive&Act driver body pose benchmark which contains frames of participants 11 to 14.

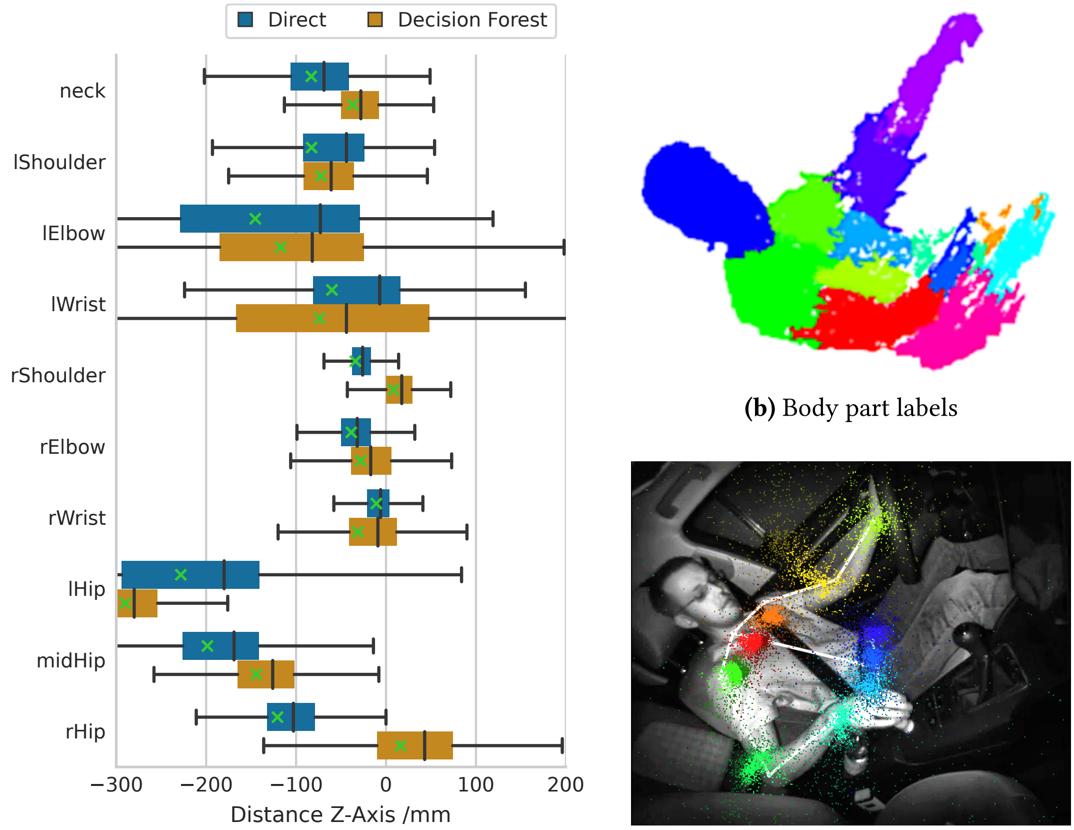
We test the real-time capabilities of our implementation on a high-end desktop system (CPU: AMD Ryzen Threadripper 1920X, GPU: nVidia 2080Ti) using CUDA 10.1 and cuDNN 7.6. We benchmark OpenPose using its default settings as it is the basis of all methods in this section. The runtime for a single frame is 28.3 ms. All other components of the system have negligible runtime. The triangulation takes about 1 ms and even the deep learning-based DepthFix method takes only 1.5 ms. The overall frame rate of all methods therefore depends on the speed of OpenPose and the number of necessary cameras. The best depth image-based method achieves 33 fps and can utilize the full frame rate of the Kinect. A two-camera triangulation-based system achieves 17 fps and a four-camera system 8.7 fps. All methods are therefore usable for many use cases needing soft real-time, like our activity recognition methods. There are other 2D body pose detectors running at higher frame rates but often with lower accuracy. All presented methods work with any generic 2D pose detector and can make use of any advancements in the area.

4.4 Evaluation on the Drive&Act Driver Body Pose Benchmark

All methods presented in this chapter were evaluated on the Drive&Act driver body pose benchmark. We first evaluate the performance of OpenPose on the different views of Drive&Act for 2D body pose estimation, followed by an analysis of the random decision forest-based method and DepthFix in comparison to the triangulation-based baseline approach. The used performance metrics are presented in Section 3.5.7. The dataset is based on a multi-view camera system. We conduct the evaluation in the camera coordinate frame of the Kinect so the x-axis and y-axis correspond to the image axes. The z-axis corresponds to the depth values of the depth image and therefore highlights the challenges for depth image-based 3D driver body pose estimation methods.

4.4.1 OpenPose Performance for 2D Driver Body Pose Estimation

Table 4.3 shows the results for 2D driver body pose estimation with OpenPose. Note, that the method was trained on the MS-Coco dataset and not on data of the interior of the car. Overall, the performance did not degrade compared to the results on the MS Coco dataset. This is unexpected because of the domain shift from color images to near-infrared images as well as the shift to a different environment. The reason could be that the driver fills most of the image in our application making the detection less challenging. Nevertheless, there are large differences in quality between the camera views. The results of the camera at the A-pillar of the driver side are worst. The reason might be that this camera is the closest and exhibits the most distorted view of the driver. In addition, the arms of the driver often occlude the hips making their detection more challenging. The cameras on the co-driver side work best, likely because their view is often less obstructed. However, if the co-driver is present additional occlusion could degrade the results as previously discussed. The results on the Kinect data are by far the best because the images are the most evenly lit with the highest contrast compared to all other camera views of the dataset. In addition, the Kinect has a narrower field of view, so some challenging poses are partially out of frame and contribute less to the measured performance.



(b) Body part labels

(a) Performance on z-axis

(c) x/y-error distribution

Figure 4.9: Random decision forest-based performance on the Drive&Act driver body pose benchmark. (c) Keypoint error distribution of the x-axis and y-axis reprojected onto the image. Bright colors represent 50 % of the samples with the smallest Euclidean distance to the ground truth. The reference ground truth body pose is depicted in white.

4.4.2 Random Decision Forests Trained on Synthetic Data

We cannot fully evaluate our approach based on random decision forest on the dataset because the method does not yield results for the eye and nose keypoints. These features are too small to be captured with our body part labeling approach. We nevertheless test its

Table 4.3: OKS scores of OpenPose for 2D driver body pose detection on different camera views.

Configuration	$\mathbf{AP}_{50-95}$	AP_{50}	AP_{75}	$\mathbf{AR}_{50-95}$	AR_{50}	AR_{75}
IR 1 (Driver A-pillar)	61.9	89.1	70.4	72.7	93.3	82.4
IR 2 (Central Mirror)	61.7	86.9	70.5	73.2	92.8	82.7
IR 3 (Co-driver A-pillar)	75.7	96.8	88.5	83.1	98.5	93.3
Kinect (Co-driver A-pillar)	**81.9**	**98.6**	**93.6**	**87.6**	**99.4**	**96.2**
All cameras	69.1	92.8	79.5	79.2	96	88.7

performance on the remaining keypoints of the dataset. Accumulated performance metrics (MPJPE, PCK, availability) are therefore not comparable to the results of the other methods. The configuration tested on the benchmark is trained on the synthetic dataset. This configuration was also used for 3D driver body pose estimation on the InCarIn activity dataset.

Table 4.4 shows the results in comparison to all other approaches presented in this chapter. While the method achieved a mean per joint position error of just 57.9 mm on the Driver Depth Pose dataset the error increases to 172.4 mm on the new benchmark. There are multiple reasons for this. The new benchmark tests the distance to the true keypoint position instead of the keypoint position on the surface of the body. Figure 4.9a highlights this. It depicts the error on the z-axis which is based on the values in the depth image. Our method based on random decision forests determines keypoints on the surface of the body, compared to the ground truth the results are therefore shifted closer to the camera causing a systematic increase of the overall error. The *direct* method that uses OpenPose results and reconstructs 3D keypoints via lookup in the depth image (see Equation 4.9) shows the same behavior with, for some keypoints, similar mean deviations.

The biggest drawback of the method is its need for a segmentation of the driver which fails often on the dataset for various reasons. While the Kinect produces less noise compared to the Melexis sensor used to record the Driver Depth Pose dataset, its depth values of surfaces shift if other surfaces come close, for example if the left arm of the driver gets close to the driver door. In addition, the large center screen acts as a mirror causing additional shifts in the depth image. Both sensor artifacts cause errors when creating the segmentation of the driver using the fixed background model. Finally, some frames of the dataset depict the driver interacting with large objects, like a newspaper, which are then part of the foreground. However, the random decision forest does not have a separate label for this, so these areas are incorrectly labeled as a body part, degrading the result. This is also reflected in Figure 4.9c where, for example, the left elbow is not well clustered but spread out because the segmentation includes the driver door in many cases.

However, while the performance is much worse, compared to the results from the Driver Depth Pose dataset, this cannot be attributed to the synthetic training data for body part labeling. If the segmentation works, the body parts are also well labeled (see Figure 4.9b).

4.4.3 DepthFix Performance

This section discusses the performance of the *DepthFix* method in comparison to the triangulation-based baseline approach. For triangulation we test different multi-view setups based on the four cameras at the A-pillar on the driver side (IR 1), central mirror (IR 2) and A-pillar on the co-driver side (IR 3 and Kinect). This includes many two-camera setups

Table 4.4: Overall results for 3D driver body pose estimation. Decision Forest results are incomplete. Values marked italic cannot be compared directly. Bold values mark the best result for multi-view and depth image-based results. The columns on the left show statistics per keypoint while the columns on the right show statistics per body pose.

Configuration	lEye	rEye	nose	lShoulder	rShoulder	neck	lElbow	rElbow	lWrist	rWrist	lHip	rHip	midHip	mpjpe	PCK_{1-15}	avail.
	mpjpe [mm]													[mm]	[%]	[%]
Multi-View																
IR3 + Kinect	37	34	37	148	76	93	197	65	128	55	165	154	142	103.3	47.7	**93.4**
IR23	19	17	23	54	38	68	74	24	66	31	122	80	92	54.5	59.7	85.1
IR12	10	**12**	16	28	41	**49**	40	32	29	31	98	116	93	45.3	59.7	77.2
IR13	14	15	19	34	28	51	**37**	32	29	27	86	**72**	70	40.4	62.5	81.3
IR123 + Kinect	10	13	**15**	30	25	56	38	23	33	22	**81**	76	**68**	37.4	**66.8**	85.3
IR123	**9**	**12**	**15**	**26**	**23**	53	**37**	**21**	**28**	**21**	82	76	**68**	**34.9**	64.7	80.2
Depth Image																
*Decision Forest**	-	-	-	129	79	117	210	107	219	133	326	240	163	*172.4*	*23.8*	*97.6*
Direct	34	30	25	103	52	120	158	54	104	38	234	138	205	100.2	48.9	97.1
Fixed Offset	25	**18**	23	78	35	78	132	**39**	103	39	121	100	93	68.8	60.7	97.1
Full Regression	19	21	23	46	33	60	63	46	71	47	91	**86**	76	55.1	64.1	97.1
DepthFix	**17**	**18**	**22**	**45**	**32**	**59**	**60**	41	**66**	**42**	**90**	87	**75**	**52.8**	**65.2**	97.1

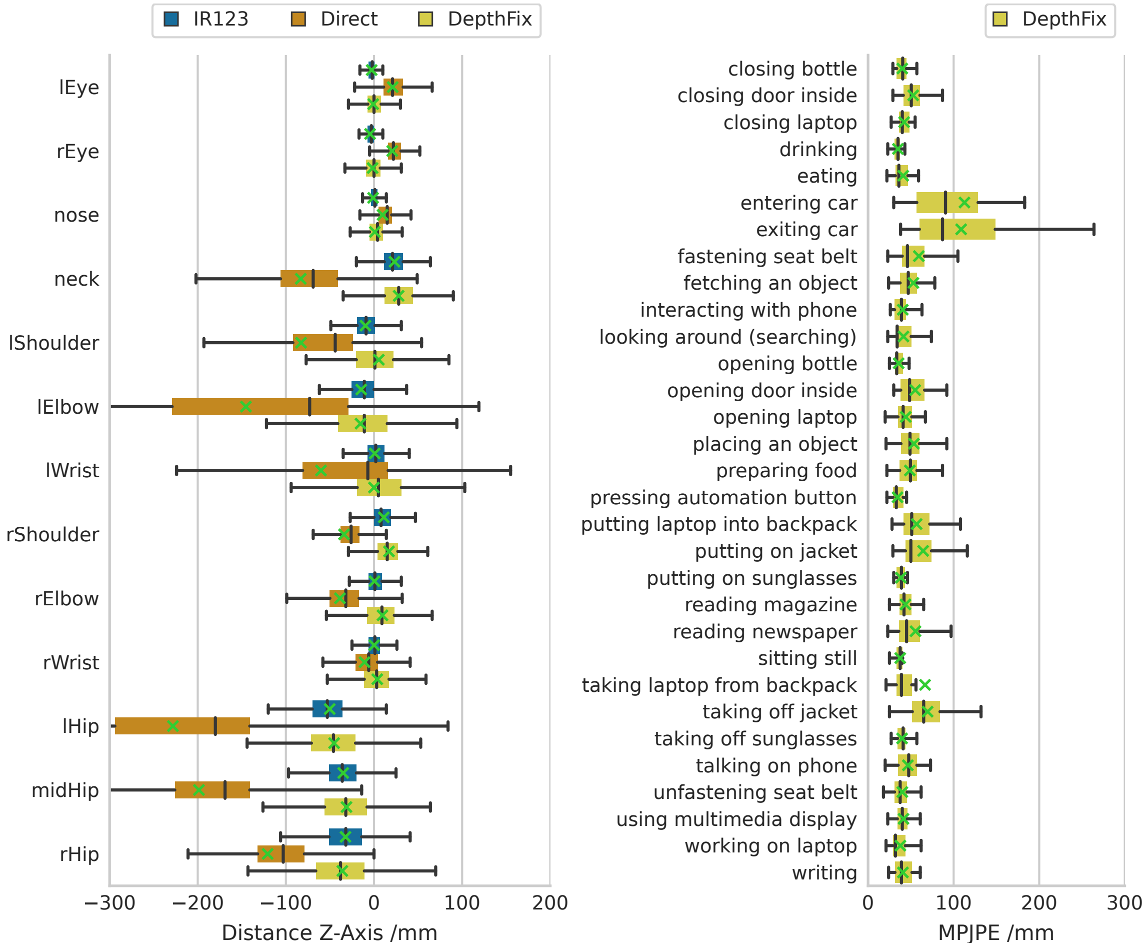
testing the impact of different camera baselines and viewpoints. In addition, we test the combination of all three near-infrared cameras (IR 1, 2, 3). This was the setup used to generate automated 3D driver body pose labels for the Drive&Act dataset. For *DepthFix* we compare to the different simpler approaches to generate 3D driver body pose data without additional machine learning using *direct* lookup (see Equation 4.9) or *fixed offsets* and the deep learning based alternative, regressing the whole 3D body pose (*full regression*).

Table 4.4 shows the results. The accuracy of keypoints on the left side of the body as well as the hips is worse compared to the right side. This is caused by the worse visibility of these keypoints because of self-occlusion and occlusion by the interior of the car. This effect is less pronounced if the camera on the left side of the body (IR 1) is part of the setup because it provides unobstructed data even if the left side is occluded in most other views.

For the triangulation-based methods the location and number of cameras makes a large difference. The combination of cameras on the co-driver side (IR 3 + Kinect) achieves the worst accuracy, however with the highest availability. The availability can be attributed to the good performance of OpenPose on these views. The low accuracy is caused by the small baseline compared to all other tested setups combined with the largest distance of these sensors to the driver. This results in large deviations of the triangulated result even for small errors of the OpenPose data. The results also show the importance of observing the driver from different sides to achieve high accuracy. Including just cameras at the center mirror and co-driver A-pillar (IR 2,3) works worse than setups including the camera at the A-pillar on the driver side (IR 1,2). The best performing two camera setup uses the cameras on both A-pillars (IR 1, 3) combining views from both sides of the driver with the widest possible baseline. The most accurate triangulation-based setup in terms of mean per joint position error (MPJPE) combines all three frontal IR cameras. However, this is a trade-off as the setup with all four cameras achieves better performance in terms of the percentage of correct keypoints metric (PCK) which considers both the position error as well as the availability of keypoints.

The results of the depth image-based methods follow the reasoning presented while introducing *DepthFix*. The *direct* approach performs worst, consistently having larger errors than the method using *fixed offsets* to move surface keypoints to their true position. Applying the two deep learning-based methods further improves the results. As expected, regressing the full 3D body pose (*full regression*) performs worse than regressing shorter offsets for each individual input keypoint with the *DepthFix* approach. However, compared to the method using *fixed offsets* the deep learning-based methods improve results mainly on keypoints that are often occluded, like the hips and the left side of the body. This highlights the effectiveness of these methods to deal with occlusions. The overall performance of *DepthFix* surpasses the triangulation-based approaches using just two cameras and reaches the quality of the best triangulation setups according to the PCK metric. However, there are trade-offs: While the best triangulation-based method produces keypoints that are more accurate (MPJPE) *DepthFix* achieves higher availability and detects overall more keypoints with less accuracy (PCK).

Figure 4.10a further investigates the accuracy of the best triangulation method compared to the *direct* depth method and *DepthFix*. The graph depicts the spread of errors along the z-axis which corresponds to the data in the depth image. It highlights the challenges for depth image-based driver body pose estimation methods (see Section 4.1). The triangulation-based method (IR 1, 2, 3) produces overall small errors that are centered around zero, indicating no bias in any direction on this axis. The hips are the exception and are on average estimated five centimeters closer to the camera. Keypoints on the left and right side of the body perform similarly which highlights the robustness of the multi-view system to occlusions. The depth image-based methods on the other hand exhibit larger errors especially for the left side of the body which is occluded more often from the viewpoint of the Kinect. The *direct* depth estimation method uses the values of the depth image directly in combination with the OpenPose results. It therefore produces keypoints on the surface of the body which biases the results towards the camera. In addition, the method produces large errors for occluded keypoints as shown by the large spread of values for keypoints on the left side of the body and the hips. The *DepthFix* method uses the result of the *direct* method as input and produces corrected results. The graph shows that *DepthFix* can correct the general bias and can estimate keypoint that are located close to their true position instead of on the surface measured by the depth image. In addition, the approach can strongly reduce the spread of errors for the left side of the body indicating that the method is able to fix large outliers caused by occlusions. The mean error of the *DepthFix* method is comparable to the triangulation-based approach, including the offset of the hip keypoints. This indicates that the *DepthFix* methods has

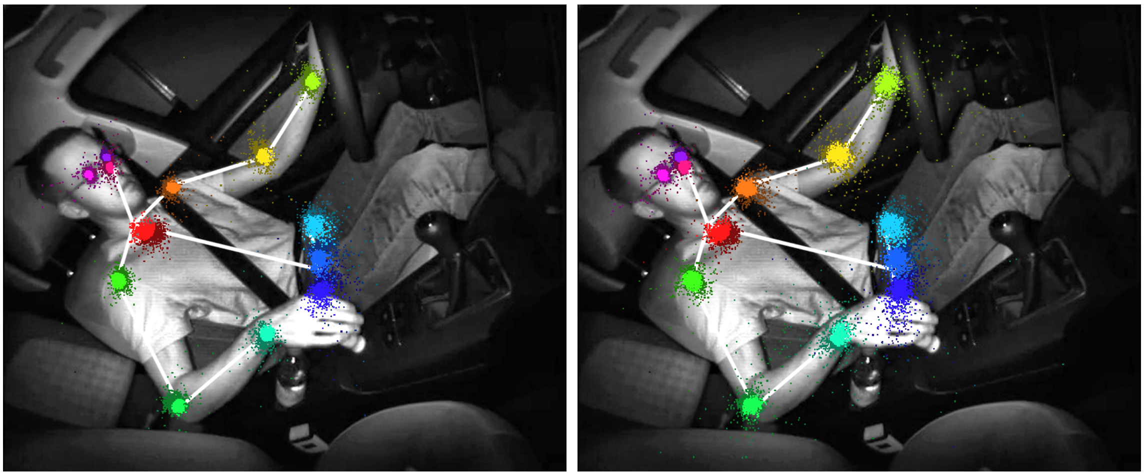


(a) Performance on z-axis

(b) Performance per activities

Figure 4.10: DepthFix performance on the Drive&Act driver body pose benchmark.

learned these biases from the triangulation method because we used its result as ground truth to train the model. The bias therefore originates from OpenPose. We visualize this effect in Figure 4.11. It shows the distribution of errors on the x-axis and y-axis projected on the image of the Kinect. Compared to the ground truth, OpenPose, as well as all our methods that rely on it, estimate the hip keypoints biased towards the knees of the driver. This position is also closer to the camera which is why the error on the z-axis discussed before showed the same effect. Overall, the distribution of errors is visually similar for both triangulation and *DepthFix* approaches. This is expected as both rely mainly on OpenPose to determine the x and y-coordinates in this camera view. However, the spread of errors of *DepthFix* is worse for the left side of the body because of the higher level of occlusions from the point of view of the Kinect. This does not affect the triangulation-based baseline because of the camera at the A-pillar on the driver side.

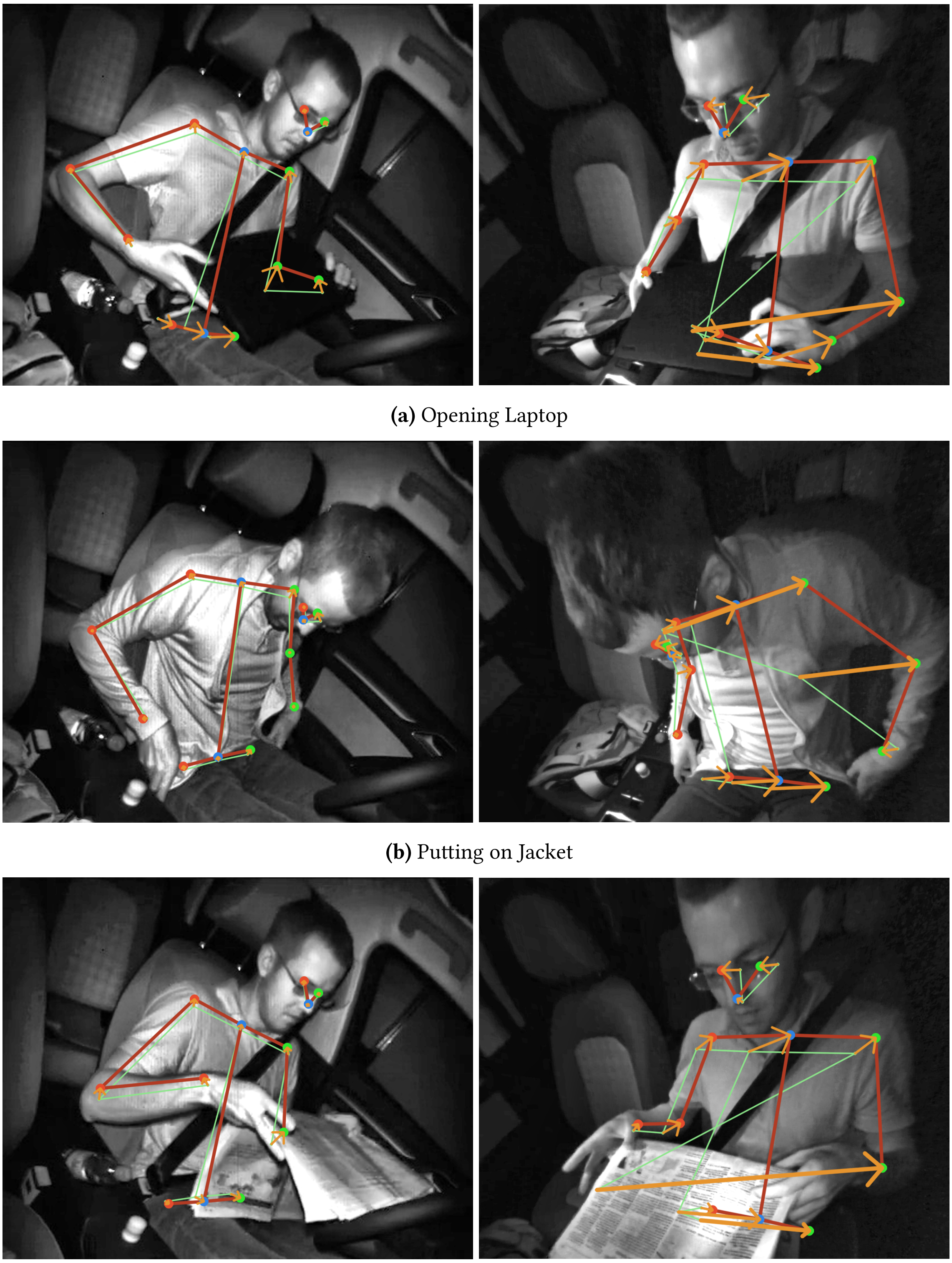


(a) NIR123 triangulation **(b)** DepthFix

Figure 4.11: x/y-error distribution of the most accurate triangulation setup compared to *DepthFix*.

Figure 4.10b shows the accuracy of *DepthFix* for different activities of the dataset. Overall, most activities have a mean per joint position error (MPJPE) of around 5 cm which corresponds to the overall result on the dataset. The activities *exiting car* and *entering car* show the largest errors. However, in these cases the driver is still (partially) outside of the car and severely occluded by themselves or the interior. Other activities with lower accuracy involve either large body movements (e.g., *fastening seat belt*), occluding objects (e.g., *putting laptop into backpack*) or both (e.g., *taking of jacket*).

Figure 4.12 shows example results of the *DepthFix* approach. The left image shows the view of the Kinect that was used to generate the result. The right image shows a projection of the 3D result onto the view of the camera at the A-pillar of the driver (IR 1). The x-axis of this view closely resembles the z-axis of the Kinect camera coordinate system. It highlights the correcting offsets estimated by *DepthFix* (orange). The left image demonstrates the ability of OpenPose to correctly estimate even occluded keypoints like the left arm, left shoulder or hips. It also shows that *DepthFix* learned to only correct the values of the x-axis and y-axis by a small amount and to rely mostly on the good accuracy of the OpenPose-based input data. The right images show the ability of *DepthFix* to correct the z-axis that is based on the data from the depth image. The method can move visible keypoints from the surface of the body to their true position (short offsets) and it can correct the position of occluded keypoints (long offsets).



(a) Opening Laptop

(b) Putting on Jacket

(c) Reading Newspaper

Figure 4.12: Results of *DepthFix* (red) for different actions. Left column shows the view of the Kinect sensor used for detection. Right column shows the re-projection of the result onto the camera image of IR 3 highlighting the correcting offsets of *DepthFix* (orange) starting from the *direct* pose input (green).

4.5 Summary

This chapter focused on 3D driver body pose estimation from depth images with real-time capable approaches. Using depth images creates unique challenges that we explored in our work: The depth image depicts only the surface of the driver from the point of view of the camera while keypoints are located inside of the body and it does not contain valid data for occluded keypoints. Furthermore, in the automotive context there were no large datasets to train methods based on machine learning.

Our first approach focused on real-time capability and the lack of training data. At that time real-time performance of deep learning-based methods was still hard to achieve. Therefore, the method relied on a segmentation of the driver and dense body part labeling via random decision forests. To train the method we relied on synthetic data using rendered depth images. We could demonstrate good performance of this approach on the Driver Depth Pose dataset, which was our first benchmark dataset for the task, achieving a mean per joint position error of 57.9 mm. However, on the Drive&Act 3D driver body pose benchmark, which we constructed to highlight the challenges for depth image-based approaches, our method only achieved an error of 172.4 mm. This is the result of the algorithm frequently failing to achieve the necessary segmentation of the driver due to complex body poses and the use of objects creating occlusions.

Our second approach focused on handling these cases with a regression-based method using deep learning. Unfortunately this also increased the demand for large-scale datasets. To address this issue we relied on a novel split of 2D body pose estimation followed by separate 3D keypoint regression guided by the depth image. This enabled us to rely on advances in 2D body pose estimation using large scale datasets from other domains. We used the 2D body pose result combined with the depth image to compute an initial 3D body pose of the driver. However, this pose is affected by different measuring artifacts of depth images. We used this result as input to a neural network to regress correcting offsets (hence the name “DepthFix”). It can be used in combination with any depth camera and any 2D body pose detector to create accurate 3D body pose results. In addition to our depth-based contributions we also created a baseline using multi-view triangulation.

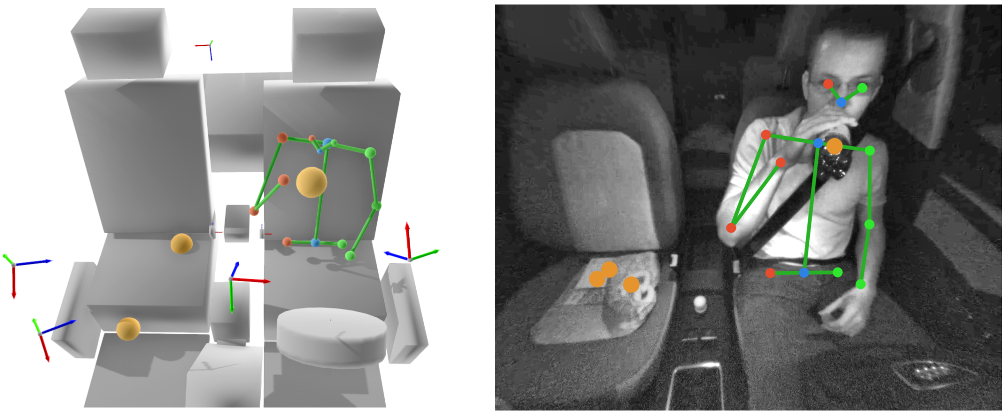
Our evaluation on the Drive&Act driver body pose benchmark showed that the method based on triangulation could achieve a mean per joint position error of 34.9 mm while DepthFix achieved 52.9 mm but with higher availability of the keypoints. Compared to our previous approach, our second method could achieve better results with less limitations on a more complex dataset. We could also demonstrate its robustness to partial occlusion which was one of our design goals. All three methods presented in this chapter were used to create 3D driver body poses for our activity recognition datasets to evaluate the methods presented in the next chapter.

5 Driver Activity Recognition

Driver activity recognition is the final goal of this thesis. Our methods rely on the datasets presented in Chapter 3 as well as the 3D driver body pose estimation methods presented in the last chapter. Compared to related approaches for 3D human body pose-based activity recognition our methods expand the input to a complex interior state model of the car including interior elements, like the steering wheel, and 3D object locations in addition to the 3D body pose of the driver (see Figure 5.1).

We investigate one central paradigm with all our methods: We assume that objects as well as interior elements are more relevant if their distance to any part of the driver is low. For example, this is the case when driving a car manually holding the steering wheel (i.e., interacting with the interior), or while drinking out of a bottle (i.e., interacting with an object). We focus on modeling this relationship and testing its influence on the performance of 3D human body pose-based activity recognition methods.

Our approaches build on top of each other. We first combine the 3D body pose of the driver with the location of interior elements to detect interactions with the interior (e.g., hands on wheel detection). We then use the same concept for activity recognition with recurrent networks. Finally, we cast the full 3D interior state model, including interior elements, objects as well as the body pose of the driver, into a spatio-temporal graph for graph convolution-based activity recognition.



(a) Activity recognition input **(b)** Reference Camera Image

Figure 5.1: (a) Interior state used as input for our activity recognition systems with the 3D body pose of the driver (green), object positions (orange) and interior elements (gray). (b) Reference camera image not part of the input.

5.1 Interior Interaction Detection

The purpose of our method for detecting interaction with the interior was to test the general concept of this chapter regarding the usefulness of modeling the distance of drivers to their surroundings (i.e., the interior elements in this experiment). The method relies on the data from our pilot study on interior interaction detection in the InCarIn project (see Section 3.4.2). This section is based on our publication at the International Conference on Intelligent Transportation Systems (ITSC) in 2017 [Mar17a] © IEEE, 2017.

5.1.1 Method

The interior state our method relies on includes the 3D driver body pose as well as the interior elements the driver interacts with in the InCarIn interaction dataset. These interior elements are modeled as geometric primitives. The dataset includes activities like *grasping the steering wheel* or *reaching to the glove box*.

Before using machine learning methods to detect complex activities in the following chapters, this section highlights the power of our input representation using an analytical approach. It detects the interaction with interior elements by thresholding the distance of keypoints of the driver's body to interior elements of interest.

To achieve this, we define a distance-based feature matrix D that represents the Euclidean distance from the position p of all body keypoint $k \in K$ of the driver to the surface s of all interior elements $i \in I$ of interest:

$$D_{k,i} = [d(p_k, s_i)] \tag{5.1}$$

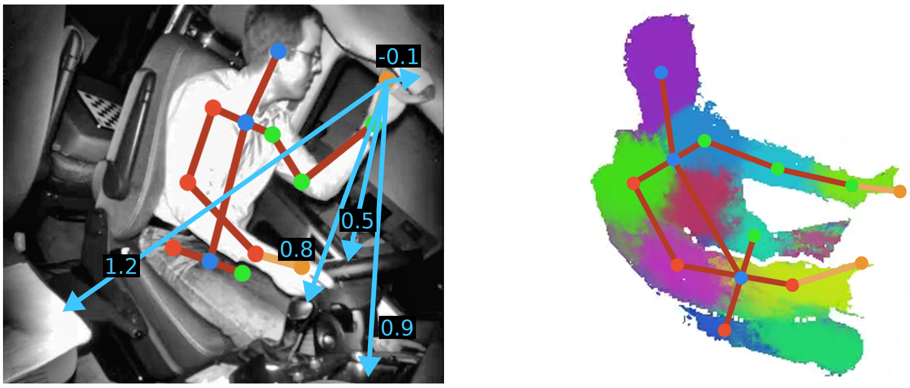
The distance function $d()$ determines the signed Euclidean distance from the position of a keypoint to the closest surface of interior elements. Depending on the type of primitive representing the interior element (e.g., cube vs. cylinder), the distance function changes accordingly. The result is negative if a keypoint lies inside of the volume.

Figure 5.2a depicts a sample of the distance feature for the left hand of the driver.

To determine if keypoint k interacts with an element we threshold feature D with a user defined threshold $t_{interaction}$:

$$M(k) = \begin{cases} i, & \exists_{=1} i \in I(D_{k,i} < t_{interaction}) \\ no\ interaction, & other \end{cases} \tag{5.2}$$

where $M(k)$ is the result of this thresholding operation for keypoint k.



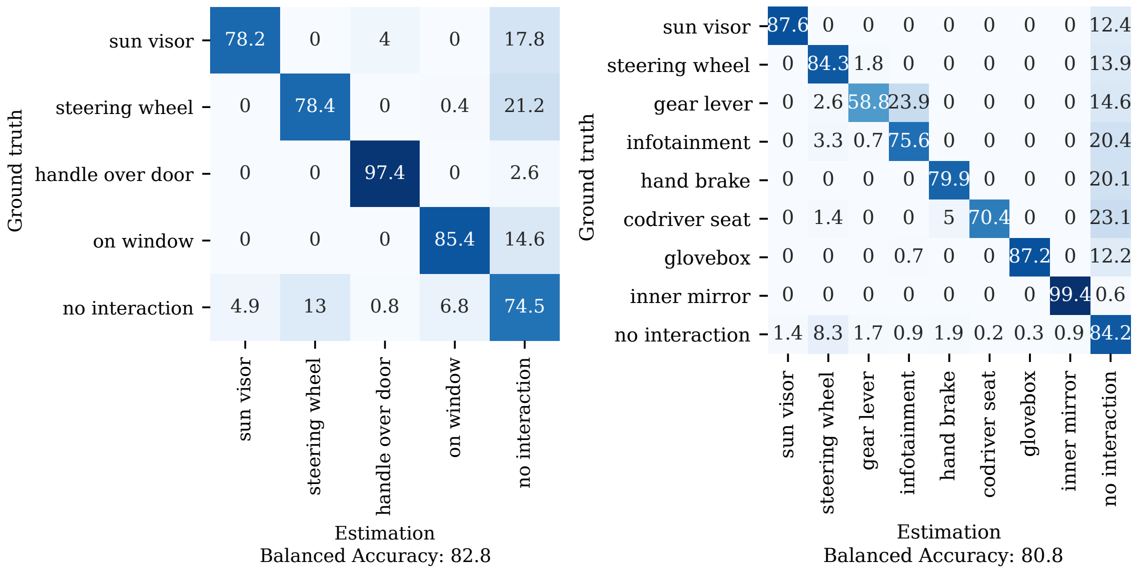
(a) Distance Feature

(b) Hand-tip estimation

Figure 5.2: (a) Depiction of the distance feature for the left *hand-tip*. The left hand grabs the handle (distance: -0.1). (b) Estimation of *hand-tip* keypoints (orange) as points furthest away from the lower arm within the cluster of the wrist.

The result is the element the keypoint interacts with if there is exactly one interior element with a distance smaller than $t_{interaction}$ to this keypoint. If the distance to all interior elements is greater than the threshold there is *no interaction*. A distance smaller than the threshold for more than one element indicates overlapping regions making the result ambiguous. This is also classified as *no interaction*. With this approach we can determine the element the driver interacts with for each keypoint individually. We can, for example, differentiate between an interaction with the left or right hand or both hands.

Our approach relies solely on the distance between the driver and interior elements. The hand pose is not part of the input. Therefore, the method cannot discern between being close to the area or interacting with the area (e.g., hand close to steering wheel vs. grasping it). Detecting the hand pose or even discrete states of the hand (e.g., grasping, pointing) is outside of the scope of this thesis. In addition, our 3D driver body pose methods only determine the position of the wrist while interactions with the surroundings usually involves the hands or fingers. Therefore, we make use of the segmentation and body part labels of our approach based on random decision forests to infer two additional keypoints on the hands closer to the location where the interaction takes place. We call these keypoints *hand-tips*. To infer their position, we combine the driver body pose result with an additional analysis of the clusters used to infer the wrist positions. We determine the direction from elbow to wrist keypoint and then compute the point of the wrist cluster that is furthest away from the wrist keypoint in this direction (see Figure 5.2b orange). This results in two additional keypoints that are located on the hand or fingers closer to where the driver interacts with interior elements or objects. However, these keypoints do not describe any specific point on the hand because their position changes depending on the hand pose and accuracy of the segmentation.



(a) Left hand

(b) Right hand

Figure 5.3: Performance of the interior interaction method for the left and right hand.

5.1.2 Evaluation on the InCarIn Interior Interaction Dataset

To evaluate our interior interaction method, we use the InCarIn Interior Interaction dataset (see Section 3.4.2) and its model of interior elements. We determine the body pose using our decision forest-based method (see Section 4.2) trained on synthetic data and extended with our approach for *hand-tip* estimation. We are only interested in the interaction of the *hand-tips* with the interior elements drivers interacted with in the dataset. Accordingly, we determine the distance feature only for *hand-tip* keypoints as well as the interior elements listed in Figure 5.3. We set the threshold for interaction detection $t_{interaction}$ to 5 cm which was the average keypoint position error of the body pose estimator on the Driver Depth Pose dataset.

Figure 5.3 shows the results for both hands. Overall, the method achieves a balanced accuracy score of 81.5 %. The performance of the system for the left hand is 1.3 p.p. (percentage points) higher while the score for the right hand is 0.7 p.p. lower. However, there are fewer regions the driver can reach with the left hand instead of the right hand because most controls are on the right side of the driver. All frames where the *hand-tip* could not be detected or where the distance to all elements is higher than the chosen threshold are labeled as *no interaction*. In addition, we assume that interaction is mutually exclusive so frames where multiple interior elements are identified for interaction are also labeled as *no interaction*. This is therefore the class that is most often confused with other areas of the interior. However, the amount of confusion with the class *no interaction* depends directly on the chosen threshold. For a threshold of zero or even a negative threshold (*hand-tip* must be inside of the interior element) most frames of the dataset would be classified as *no interaction*. For small thresholds, the method is therefore unlikely to confuse interactions with elements that are far from each other if the body pose is detected correctly. This holds true for most areas as shown in the confusion matrices. Areas that are close to each other can be confused more easily depending on the accuracy of the body pose detector and the chosen threshold. This is for example the case for the areas *gear lever* and *infotainment* of the right hand because the gear lever in the VW-T5 test vehicle is part of the dashboard and integrated in the infotainment area.

[Ohn13] implemented a similar application using a color camera but classifying only a small set of regions like the steering wheel, gear lever and infotainment area. They used a camera at the ceiling with an optimal view of the dashboard, compared to the side-view of our setup. Their method relies on the definition of regions of interest in the image and training of support vector machines (SVM) for each region to detect the presence of one or both hands. While it is hard to compare the methods without an evaluation on the same data, our approach reaches similar performance on more regions that we can freely define without a training process. Furthermore, the pose estimation algorithm that our system is based on is not limited to this application. In addition, our method for interior interaction detection adds very little processing time (<1 ms) to the base system compared to an approach based on machine learning.

The results show that our method works well in general and that the distance from the driver to the surroundings is, at least in the tested scenario, very distinctive. In addition, the approach is easily extendable to additional body parts and areas of interest by changing the configuration and extending the list of interior elements. Distance closely correlates with the intent to interact with interior elements. For example, *reaching for the infotainment area* or *the co-driver seat* indicates intent to interact with objects in that area. However, there are other regions where monitoring the distance might not indicate an immediate activity. Some people, for example, rest their hand on or near the gear lever without the intention to use it. Similarly, the hands can be close to the steering wheel while resting on the legs of the driver. Therefore, an optical hands-on wheel detection system that should replace current sensor based methods would need to detect if the driver is grasping the steering wheel. However, this would increase the complexity and necessary computing power by a large margin.

5.2 Three-Stream Recurrent Driver Activity Recognition

Our method for detecting interaction between the driver and interior elements demonstrated that our concept of relying on the distance from 3D driver body keypoints to surround elements works at least for activities with a simple correlation between these elements. We therefore apply the same concept for detecting more complex secondary activities like eating, drinking, or talking on the phone. To this end we take the distance feature, that was at the core of our last approach (see Equation 5.1), and investigate its usefulness as input for a deep learning-based driver activity recognition method. While our first method relied on data from a single frame, we also consider the temporal sequence of many complex activities by using recurrent neural networks for our approach.

In the following we first give a brief overview of recurrent neural networks in general before describing our approach for driver activity recognition including 3D interior elements as input feature. The following sections are based on our publication at the International Vehicles Symposium (IV) in 2018 [Mar18b] © IEEE, 2018.

5.2.1 Recurrent Networks

Time is an important factor in detecting and disambiguating activities. The Drive&Act dataset, for example, includes fine-grained labels like *opening* and *closing a bottle*. Considering only a single point in time these activities can look the same, while they are certainly different when observing the activity for the whole duration. Recurrent neural networks are designed to analyze such time series data.

A basic recurrent neural network layer can be defined in the following way:

$$h_t = f(Ux_t + Vh_{t-1} + b) \tag{5.3}$$

Where x_t is the input vector at time t, U and V are weight matrices, b is a bias vector, f is a nonlinear function and h_t is the output of the recurrent network layer at time t. The output of the network in the current time step therefore depends both on the current input and on the output of the network in the last time step.

These networks are often used with a fixed time horizon n of at most a few hundred steps. In the first time step of an evaluation $t = 0$ the previous output h_{-1} is unknown and is often initialized with zero. When training such a network the gradient needs to be backpropagated through all n time steps. This is the main drawback of this formulation because it requires the gradient to be retained through all steps while flowing through

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

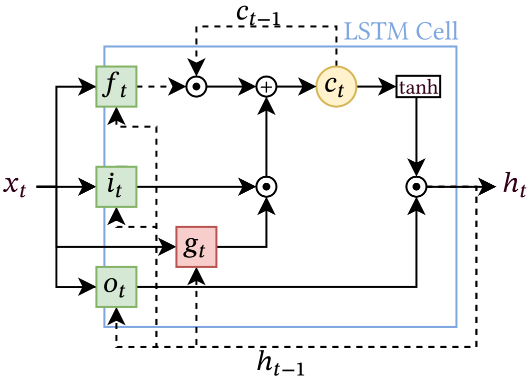


Figure 5.4: LSTM cell definition for time step t. i, f, o are the input, forget and output gates of the cell. g is the cell update function. c is the state of the cell and h is the hidden output state. x is the input feature vector. W and b are the trainable weights and biases. σ is the sigmoid function and $\odot$ indicates element-wise multiplication. Dashed lines indicate recurrent connections.

all parts of each layer. This can limit the length of the input sequence and can lead to gradients that either vanish or blow-up making training unstable.

For this reason, most activity recognition methods that rely on recurrent neural networks use layers with long short term memory cells (LSTM) because they alleviate these challenges [Hoc97]. They achieve this by introducing a memory cell with state vector c and by moderating the flow of information from and to this memory cell using gates. Each gate and the memory cell itself are modeled as feed forward layers. Figure 5.4 shows the definition of an LSTM cell according to Hochreiter et al. [Hoc97] and according to the implementation in pytorch[1], used throughout this thesis. This design allows the gradient to be retained in long sequences because the recurrency of the state vector c_{t-1} acts as a shortcut with only a few primitive operations that do not degrade the gradient compared to the simple formulation. All three gates (i, f, o) as well as the cell state update function (g) depend on the current feature vector x_t as well as the previous hidden state h_{t-1}. Each gate determines weights that are used for different functions within the LSTM cell. The input gate i determines the importance of the input feature x_t to moderate the update of the cell state. The forget gate f determines the importance of the last cell state c_{t-1} and allows the cell to forget data that is no longer relevant. Finally, the output gate o determines the importance of the current cell state c_t and regulates the weight of the cell output h_t at time t.

Like other neural network layers, recurrent units can be stacked into multi-layer networks. The hidden state vector h_t then acts as the input for each time step of the following layer.

[1] https://pytorch.org/docs/1.4.0/nn.html?highlight=lstm#torch.nn.LSTM, accessed: July 19, 2022

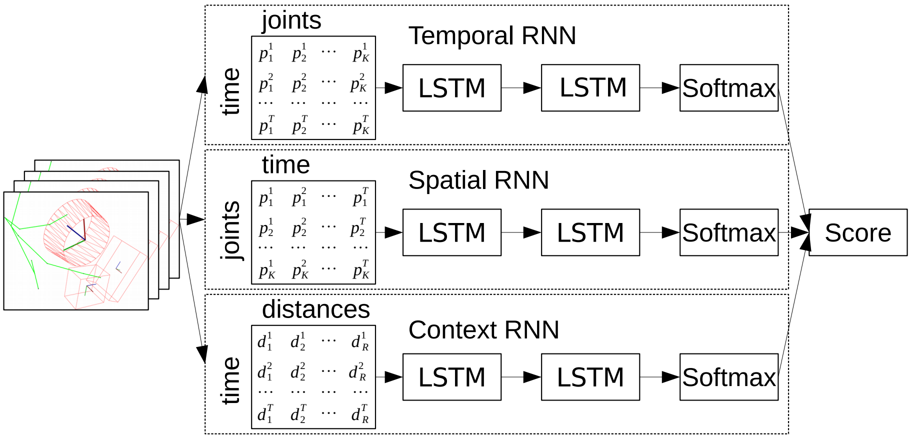

joints
Temporal RNN
time
$p_1^1 \; p_2^1 \cdots p_K^1$
$p_1^2 \; p_2^2 \cdots p_K^2$
$p_1^T \; p_2^T \cdots p_K^T$
LSTM
LSTM
Softmax
time
Spatial RNN
joints
$p_1^1 \; p_1^2 \cdots p_1^T$
$p_2^1 \; p_2^2 \cdots p_2^T$
$p_K^1 \; p_K^2 \cdots p_K^T$
LSTM
LSTM
Softmax
Score
distances
Context RNN
time
$d_1^1 \; d_2^1 \cdots d_R^1$
$d_1^2 \; d_2^2 \cdots d_R^2$
$d_1^T \; d_2^T \cdots d_R^T$
LSTM
LSTM
Softmax

Figure 5.5: Overview of our *Three-Stream* approach for driver activity recognition.

5.2.2 Three-Stream Recurrent Neural Network

Our 3D interior state model, including the 3D body pose of the driver and interior elements, is a low dimensional representation of the spatial state of the car's interior. Changes of this interior state model over time describe the temporal evolution of activities performed by the driver. While recurrent neural networks are well suited to model the temporal evolution, exploiting the spatial relationship between keypoints of the human body as well as other elements of our state model is difficult as we showed in our discussion of related work.

Wang et al. [Wan17] tackled this challenge using recurrent networks in two streams. One stream focused on the temporal evolution of the input while the second stream focused on the spatial relationship between keypoints of the human body. We extend this framework with an additional third stream that integrates 3D interior elements (see Figure 5.5). We call this stream the context stream. It describes the movements of the driver in context to interior elements. This information should be helpful for activity recognition because interaction with the surrounding environment is part of many activities. Our previous experiment for interior interaction detection supports this thesis. For example, if the driver has both hands on the steering wheel, they cannot at the same time use a smartphone or drink out of a bottle.

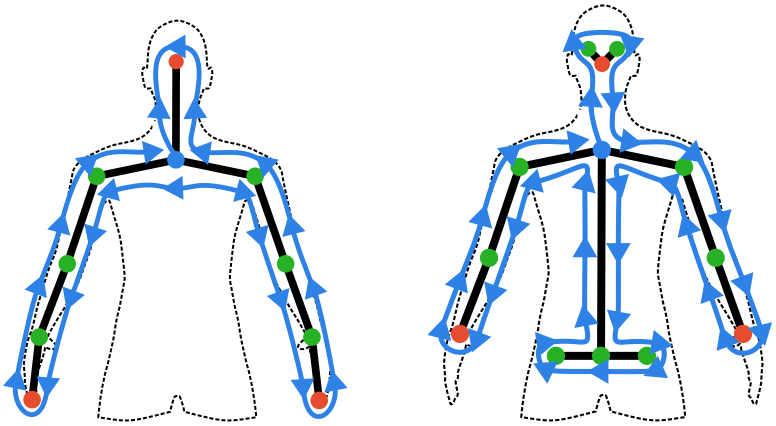
The three streams, temporal, spatial, and context, are trained separately and merged in a late fusion approach. Each stream is a separate network with the same architecture consisting of two LSTM layers followed by a layer for classification with softmax activation. In the following we present the input of each stream in detail as well as the method to combine results of different streams.

The temporal stream models the temporal dynamics within the sequences of driver body poses. It encodes the 3D keypoints $k \in K$ of the driver by concatenating them to form a stacked input vector of size $3K$ for each time step.

The spatial stream models the spatial dynamics within the human body. The idea behind this approach is that the human body consists of rigid limbs and joints that move in a fixed relationship. Depending on the performed action this results in different complex patterns. The spatial stream extracts these patterns by traversing the human body along the limbs instead of traversing the temporal domain (see Figure 5.6). Similar to Liu et al. [Liu16] we implement this using depth-first search on the kinematic tree of the human body starting at the neck. This results in a sequence of body keypoints where inner keypoints (e.g., elbows, shoulders) are visited multiple times. In this way the tree structure of the human body can be unrolled into a continuous chain without breaking the relationship between neighboring keypoints. The number of keypoints and the details of the traversal depend on the dataset. The number of steps of this stream does not depend on the temporal dimension but instead on the length of the traversal path on the body pose of the driver. The input of each step is the position of a single keypoint of the body. To still represent the temporal dimension, we concatenate the position of the current keypoint of T time steps forming an input vector of size $3T$. The temporal window of the spatial stream is equal or smaller than the temporal sequence of each sample, processed by the other streams.

The context stream models the dynamics between the keypoints of the human body and interior elements. This is the main contribution of our approach. It relies on the distance feature introduced for interior interaction detection in the last section (see Equation 5.1). The context stream allows the approach to interpret the movements of the driver within the context of the surrounding interior elements. For each time step the method determines the distance feature of K keypoints and I interior elements resulting an an input vector of size $K * I$. This feature can get large for all keypoints and interior elements compared to the input of the other streams. In addition, the distance of torso keypoints to interior elements is redundant and static most of the time because of the fixed seating position. For this reason, only a subset of the driver body keypoints are used to determine the distance feature.

Combined models use late fusion to aggregate the results of two or more streams via weighted averaging. We determine these weights by maximizing the performance of the combined model on the validation set. The combination of the temporal and the spatial stream represents the method of Wang et al. [Wan17]. We call this model *Two-Stream* in the following. The combination of the temporal and the context stream as well as the combination of all three streams is our contribution. We call the network combining all streams *Three-Stream* in the following.



(a) InCarIn Activity

(b) Drive&Act

Figure 5.6: Depiction of the body keypoints used by the method for both datasets. The temporal stream uses the keypoint positions stacked per time step. The spatial stream traverses the body pose (blue arrows) starting at the neck (blue). The interior stream uses distances from the red keypoints to interior elements (not shown).

5.2.3 Implementation Details

The approach is implemented in PyTorch 1.5.0 with CUDA 10.1 and cuDNN 7.6.5 using Ray 1.1.0 [Lia18] as distributed training framework. Each stream is trained separately using the Adam optimizer [Kin14] with default parameters and categorical cross entropy loss. Models are trained for 100 epochs using a batch size of 128 samples and 50 % Dropout after the first LSTM layer. Different streams are combined using weighted averaging. We determine the weights on the validation set using grid search.

The hyper-parameters of the approach are summarized in Table 5.1. They were determined using random search of the parameter space with Asynchronous Hyper Band Scheduling [Li20]. The training parameters were optimized on the InCarIn activity dataset and were reused for the Drive&Act dataset. The parameters of each stream were optimized separately for both datasets. For both datasets the number of samples varies greatly between classes, we therefore use random sampling to generate each training batch over-sampling classes with few samples to create an equal distribution between classes for training.

We normalize the skeleton data used as input for the temporal and spatial stream by subtracting the average of all valid keypoints. The common approach would be to subtract a dedicated keypoint, like the neck. However, if this keypoint is not available, normalization would not be possible. Our approach is therefore more robust. To determine the distances to interior elements, we use the body pose without normalization. The distance feature is not normalized further but because of the confined space its value range is close to zero.

Table 5.1: Hyper-parameters of the *Three-Stream* activity recogniton method for both activity datasets.

	Parameters	**InCarIn Activity**	**Drive&Act**
Dataset	# Body Keypoints	10	13
	Sample time steps	300	90
Temporal Stream	LSTM Size	256	512
Spatial Stream	LSTM Size	256	512
	Time Window	51	90
Context Stream	LSTM Size	256	512
	# Interior Elements	8	14
	# Body Keypoints	3 (hand-tips, head)	3 (wrists, nose)

The body pose annotations differ between the two datasets which has consequences for the traversal scheme of the spatial stream (see Figure 5.6). The annotations for the InCarIn activity dataset were created using our approach based on random decision forests trained on synthetic data. As the evaluation of this method showed, the hips are less accurate compared to other keypoints. We therefore leave them out for activity recognition on this dataset to reduce the noise of the input data. For Drive&Act the main body pose annotations were created using triangulation of the three frontal NIR cameras (on both A-pillars and the central mirror). As these results are much more accurate compared to results of our method based on random decision forests, we use them all as input.

To determine the distance feature of the context stream we select only a small subset of body keypoints and interior elements of the interior state model. The three keypoints selected for both datasets serve the same purpose. They include the hands, because they are the most important for most activities, and the position of the head as an indicator of whole-body movement. For example, if the driver leans forward or to the side this joint moves the most in relationship to interior elements. The number of interior elements is different for each dataset. We choose all elements provided with the InCarIn activity dataset. It was created for the purpose of detecting interactions with the interior. As such all regions are relevant for activity detection as well. For Drive&Act we choose a subset of interior elements consisting of storage areas, seats, and controls. There are additional areas, like the foot well, that we assume are less relevant and left out.

We test the runtime performance of the configuration used for Drive&Act on notebook hardware (Intel i7-9750H CPU @ 2.60GHz; GeForce GTX 1650 Max-Q) and achieve 44 fps for the temporal stream, 109 fps for the spatial stream and 42 fps for the context stream. Running all streams in sequence on the same graphics card results in 18 fps. Because most activities take some time the runtime is good enough for most real world applications.

5.2.4 Evaluation on the InCarIn Activity Dataset

We first evaluate our approach on the InCarIn activity dataset. It was the benchmarking dataset for publishing the method [Mar18b]. In addition, we also used this approach as baseline for publishing the Drive&Act dataset [Mar19]. We discuss these results in Section 5.4.2.

Figure 5.7a shows a comparison of the performance of each stream and stream combination. We determine both F1-Scores and balanced accuracy to be consistent with our first publication, which used F1-Scores, as well as with the Drive&Act benchmark, which uses the balanced accuracy metric. We present the InCarIn activity dataset in detail in Section 3.4. The different metrics are discussed in Section 3.5.7.

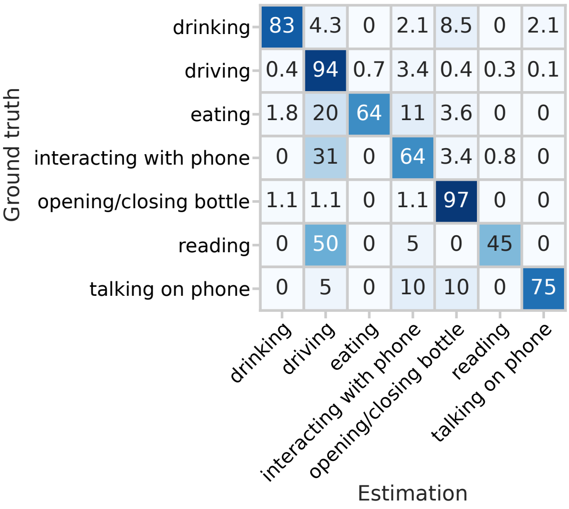
Comparing single stream performance the spatial stream, traversing the body pose, performs worst while the temporal stream, processing stacked keypoints of the driver's body for each time step, works best, with an F1-Score difference of 30.2 p.p. (percentage points). The context stream also performs worse than the temporal stream, however, the difference is only 3 p.p. (F1-Score). This indicates that keypoint positions evaluated over time are still the best descriptor for activity recognition, out of the three tested inputs. The distance feature introduced in this chapter mixes information about the driver's movement with the location of interior elements. While this reduces the performance compared to the temporal stream, it is a much better representation for activity recognition when compared to the traversal of the body pose by the spatial stream. Comparing the performance of each stream separately is only a first indicator of the combined stream performance because it does not show if streams provide complementary results.

Comparing the results of stream late fusion with the temporal stream, the performance increases overall. The combination of the temporal and spatial stream improves the results by 2.6 p.p. (F1-Score) while the combination of the temporal stream with the context stream improves results by 3.4 p.p.. Combining all three streams improves the F1-Score by 4.4 p.p.. The results indicate that the context stream provides overall more additional information compared to the spatial stream. Therefore, our addition of 3D interior elements proves to be useful for activity recognition.

The balanced accuracy metric (bal. acc.) shows the same trend with regards to the performance of single streams. For the tested stream combinations the spatial stream only improves performance when combined with the temporal stream alone. The combination including just the context stream overall perform best. Combining all three streams performs worse than combining just the context and temporal streams. Balanced accuracy focuses on the per class recall while the F1-Score is the harmonic mean of precision and recall. The difference between the metrics therefore indicates that the combination

Trial	F1-Score	Bal. Acc.
Spatial	43.6	43.1
Context	70.8	71.2
Temporal	73.8	72.5
Temporal + spatial	76.4	73.4
Temporal + context	77.2	**75.1**
Three-Stream	**78.2**	74.7

(a) Stream Performance Comparison



Ground truth \ Estimation	drinking	driving	eating	interacting with phone	opening/closing bottle	reading	talking on phone
drinking	83	4.3	0	2.1	8.5	0	2.1
driving	0.4	94	0.7	3.4	0.4	0.3	0.1
eating	1.8	20	64	11	3.6	0	0
interacting with phone	0	31	0	64	3.4	0.8	0
opening/closing bottle	1.1	1.1	0	1.1	97	0	0
reading	0	50	0	5	0	45	0
talking on phone	0	5	0	10	10	0	75

(b) *Three-Stream* Confusion Matrix

Figure 5.7: Results of our *Three-Stream* model on the InCarIn activity dataset. Best results marked bold.

of the temporal and context stream can detect more positive samples per class (balanced accuracy) but with a higher rate of false positives compared to the *Three-Stream* model.

Figure 5.7b shows the confusion matrix of the *Three-Stream* model. It shows that secondary activities are most likely confused with the much more probable driving task. The confusion between different secondary activities is low overall. The activity *reading* is the most likely to be confused with the driving activity. We can explain this with the behavior of our test participants. Most of them performed the reading task by opening the book, placing it on the steering wheel and holding both with both hands. In this scenario the body pose of the driver is the same for the reading and driving task, making activity detection based on the inputs of our model ambiguous. As we will show with our next approach and our evaluation on the Drive&Act dataset, knowledge of objects in the scene is important to disambiguate these cases.

5.3 Dynamic Interaction Graphs

Our Three-Stream approach for driver activity recognition with recurrent neural networks introduced the location of interior elements to body pose-based activity recognition. We were able show the advantages of the additional input in our evaluation on the InCarIn activity dataset. However, we also identified two shortcomings of this approach. First, the method did not perform well at capturing the spatial relationships between the keypoints of the human body as shown by the performance of the spatial stream. Second, with just the body pose and interior elements as input some activities seem similar making the recognition task ambiguous.

We address these drawbacks with our next method. It is based on graph convolutions, which are well suited for analyzing the spatial structure of the human body as it naturally forms a graph based on the human skeleton. In addition, we focus on the integration of object locations as an additional input modality to disambiguate the fine-grained activities that are part of the Drive&Act dataset. While graph convolutions are popular for body pose-based activity recognition, as we showed in our review of related methods, the addition of interior elements and object positions is a novel contribution. However, it increases the complexity of the input in the spatial domain which raises the question of how to extend the structure of the graph to include these elements in a meaningful way. The graph creation process is our main contribution. It follows the same principle as our previous methods and is guided by the distance of objects and interior elements to the body pose of the driver. We call the resulting graph *Dynamic Interaction Graph.*

In the following we first introduce graph convolution and its extensions for activity recognition in general. We then introduce our method to create *Dynamic Interaction Graphs* including objects and interior elements in addition to the body pose of the driver. Finally, we describe the neural network architecture to infer activities based on this graph structure.

The following sections are based on our publication at the International Conference on Intelligent Transportation Systems (ITSC) in 2020 [Mar17a] © IEEE, 2020.

5.3.1 Spatio-Temporal Graph Convolution

Graph convolution gained rapid popularity for human body pose-based activity recognition in recent years because it is well suited to represent the spatial relationships within the kinematic model of the human body. A popular starting point is the graph convolution definition introduced by Kipf et al. [Kip17] for semi-supervised classification of nodes in undirected graphs. There are different extensions for activity recognition extending this definition to spatio-temporal graph convolutions. We follow the approach introduced by Yan et al. [Yan18]. The following section summarizes these concepts.

Before extending the graph to the temporal domain we first introduce graph convolution on a single time step. An undirected graph consists of a pair $G = (V,E)$ with N nodes $v_i \in V$ and $(v_i, v_j) \wedge (v_j, v_i) \in E$ edges. Such a graph can also be represented by a symmetric adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N \times N}$. In addition, each node is described by a feature vector $f_v \in \mathbb{R}^K$. These can be combined to form a feature matrix $H \in \mathbb{R}^{N \times K}$. The goal of the graph convolution as defined by Kipf et al. is to compute a new feature matrix H^{l+1} per layer l based on the features of neighboring nodes in the graph defined by A and H, and a learnable weight matrix $W^l \in \mathbb{R}^{K^{(l)} \times K^{(l+1)}}$:

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{5.4}$$

$\tilde{A} = A + I_N$ is the adjacency matrix with added self-connections. $\tilde{D_{ii}} = \sum_j \tilde{A}_{ij}$ is the degree matrix of $\tilde{A}$ and is used to normalize the adjacency matrix regarding the number of connections to neighboring nodes. $\sigma()$ denotes an activation function.

This definition has the drawback that there is no order to the neighborhood defined by the adjacency matrix. The same weights are therefore used on all neighbors and the node itself resulting in a simple averaging of these nodes. In contrast to this definition convolution on images, where the neighborhood is well defined, uses different filter weights for each neighbor (e.g., nine weights for a 3 × 3 convolution).

Yan et al. [Yan18] therefore extended the graph convolution operation for activity recognition introducing a neighborhood based on the kinematic model of the human body. They define this neighborhood based on the distance of body keypoints to the center of the body. They determine the distance d to this global reference point for each node $i, j \in N$ and use it to decompose the normalized adjacency matrix into three subsets: Edges to nodes that are nearer to the reference point, edges to nodes that are the same distance – usually only the node itself, and edges to nodes that are further away:

$$s(i,j) = \begin{cases} 0, d_i = d_j \\ 1, d_i < d_j \\ 2, d_i > d_j \end{cases} \tag{5.5}$$

$s(i,j)$ denotes the partitioning function. It can be applied to the normalized adjacency matrix to get partitioned adjacency matrices $\tilde{A}_p$:

$$\tilde{A}_p = [a_{ij}^{(p)}], a_{i,j}^{(p)} = \begin{cases} a_{i,j}, p \equiv s(i,j), a_{i,j} \in \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \\ 0, \text{else} \end{cases} \tag{5.6}$$

This leads to the final definition of the spatial graph convolution operation for body pose-based activity recognition:

$$H_s^{(l+1)} = \sigma(\sum_{p \in [0..2]} ((M_p^{(l)} \odot \tilde{A}_p) H^{(l)} W_p^{(l)}) \tag{5.7}$$

Like Yan et al. [Yan18] we add an additional learnable edge weighting matrix M. $\odot$ denotes element wise multiplication. M is initialized to one. It enables the network to increase or decrease the relevance of certain edges. Using multiplication limits the network to regularizing edges that are already defined by the adjacency matrices. As we showed in our discussion of related work there are different ways of modifying the initial adjacency matrix to allow the graph convolution layer to adapt the graph in the training process. For example, using addition instead of element wise multiplication enables the network to create additional edges. However, this leads to fully connected graphs with varying edge weights that are hard to interpret. The advantage of our graph creation process, as explained in the following, is its sparseness which allows us to analyze and understand the graph visually.

We expand the introduced spatial graph convolution to the temporal domain T by adding an additional dimension to the feature matrix $H_t \in \mathbb{R}^{T \times N \times K}$. Equation 5.7 stays unchanged and is applied to each time step $t \in T$. To connect time steps with each other we add edges connecting the same node over all time steps. This can be represented by a standard convolution with kernel size $W_t \in \mathbb{R}^{F \times 1 \times K_s^{(l+1)}}$ along the temporal axis of the result of the spatial graph convolution:

$$H_t^{(l+1)} = H_s^{l+1} * W_t \tag{5.8}$$

H_t is the result of the spatio-temporal graph convolution. F is the kernel size in the temporal dimension. $*$ denotes the convolution operation.

This spatio-temporal convolution layer has two hyper-parameters, the size of the output node feature of the spatial graph convolution $K_s^{(l+1)}$ and the size of the kernel of the temporal convolution F.

5.3.2 Dynamic Interaction Graphs for Activity Recognition

Related methods for 3D body pose-based activity recognition usually only use the 3D human body pose as input feature. However, this means that graph convolution-based methods for this task use a graph with a fixed number of nodes corresponding to the number of keypoints of the human body. In our case there are both fixed interior elements that are always there and do not change over time and in addition there are objects that can move and may or may not be there. In addition, both interior elements and objects might not be relevant for the current activity even if they are visible. To keep the complexity of the graph low, making it interpretable even by a human observer, we try to keep the graph sparse. We follow the same principle as in the rest of this chapter and construct the graph based on the distance of the driver's body pose to other elements. However, we no longer rely on the distance feature used for the last two methods (see Equation 5.1) but use this concept to determine the edges of the Dynamic Interaction Graph. With the introduction of object positions in addition to interior elements the relevance of this concept increases further because activities that involve objects usually lead to the relevant objects being close to the driver. Our approach for graph construction is an analytic method and is not based on a learning process. However, the edges and nodes depend on the current input sample consisting of T time steps where each time step includes the body pose of

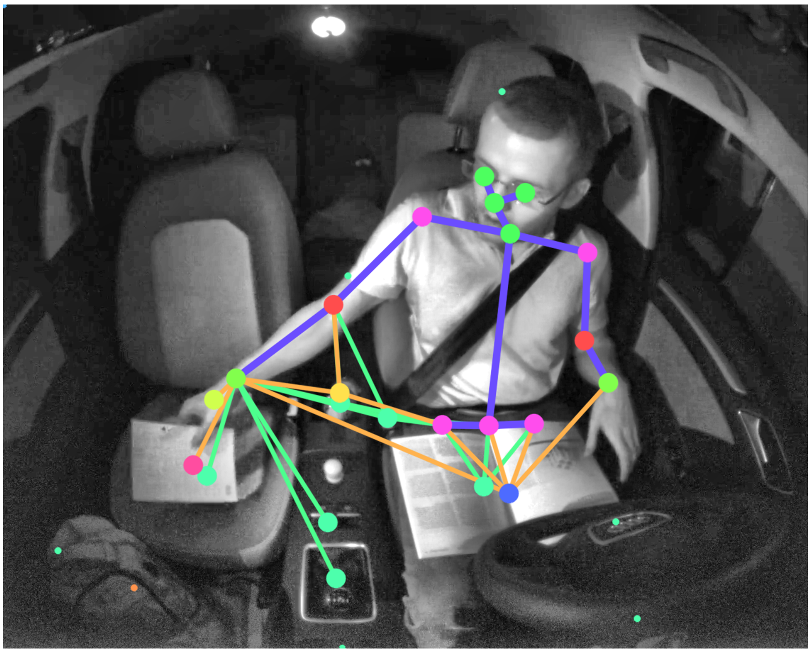


Figure 5.8: Reprojection of the 3D Dynamic Interaction Graph on a reference image for the activity *fetching an object* with body pose edges (blue), interior edges (orange) and object edges (green). The color of each point indicates its category.

the driver, the location of interior elements and of objects. Each graph convolution layer can modify the importance of the created edges in the training process (see Equation 5.7).

In the following we describe the creation process of the interaction graph in detail. It consists of three parts: The feature vector f_v of each node, the selection process of nodes V that are part of the graph and the creation of edges E between nodes. Figure 5.8 shows a sample of the resulting graph.

The node features f_v consist of the 3D position in world coordinates of the respective node V in the graph: $f_v = [x,y,z] \in \mathbb{R}^3$. Interior elements have a fixed location so their position is duplicated to all time steps of the graph. Compared to previous methods in this chapter we use the center of the interior element instead of points on the surface. Objects and keypoints of the driver's body may not be available for all time steps of the sample window. Missing observations are therefore filled with zeros.

The 3D position is the main part of the node feature. However, neither this feature nor the structure of the graph indicates the type of the node. The neural network therefore cannot easily discern between graph nodes representing body keypoints, objects or interior elements. We therefore extend the node feature with an additional category descriptor which is a one-hot encoded vector indicating the type of the node (e.g., elbow, shoulder, object). The granularity of the category vector is part of the implementation details.

All nodes V belong to the set $V \subseteq \{V_{\text{pose}}, V_{\text{interior}}, V_{\text{objects}}\}$ consisting of the three input modalities. However, only nodes connected with edges $V = \{v_i | (v_i, v_j) \in E\}$ are part of the graph even if they are available. Interior elements, for example, are always there but are not part of the graph if they are not connected. The number of nodes in the final graph therefore depends on the edge creation process.

The edges E of the graph guide the whole graph creation process. To identify suitable edges, we consider all available input elements in the time window of the sample as graph nodes.

There are two types of edges $E = \{E_{\text{pose}}, E_{\text{interaction}}\}$. Like many approaches that use just the human body pose for activity recognition, we define fixed edges E_{pose} that resemble the kinematic model of the body (see Figure 5.8 blue). Connecting interior and object nodes to this fixed graph is our primary focus. As explained above our goal is to add edges $E_{\text{interaction}}$ and therefore nodes that are relevant for the current activity. We formulate this by using the Euclidean distance between the keypoints of the body and other elements. However, if the driver is moving the distances change for each time step t of the sampling interval T. We therefore use the minimum Euclidean distance between node features f_v over the sampling interval to construct the graph. For example, for the action *picking up an object* the distance of the wrist, while moving towards the object, will be large but it will be low at the end of the action. Our graph creation process can represent these cases well (see Figure 5.8 green/orange). Formally $E_{\text{interaction}}$ is defined as follows:

$$\begin{aligned} E_{\text{interaction}} = \{(v_i, v_j) | v_i \in V_{\text{pose}} \wedge v_j \in \{V_{\text{interior}}, V_{\text{objects}}\} \wedge \\ \min_{t \in T} ||f_i^t - f_j^t||_2 < \theta\} \end{aligned} \tag{5.9}$$

θ is the threshold that regulates how many edges are created and therefore also which nodes are added to the final graph. If θ is negative no interaction edges will be created. The graph would just contain E_{pose} edges and V_{pose} nodes. On the other hand, with θ greater than the largest distance between nodes V_{pose} and other elements the graph would be fully connected between all keypoints of the driver's body and all other elements. We evaluate the impact of this threshold in Section 5.4.1.

The edge creation approach is similar to our thresholding-based method for interior interaction detection (see Equation 5.2). However, instead of using thresholding to arrive at the final decision, the edge creation process only determines the layout of the graph leaving the final decision to the graph convolution network. It can therefore also act as a pre-filtering step to remove unnecessary nodes from the graph, by not creating any edges for that node.

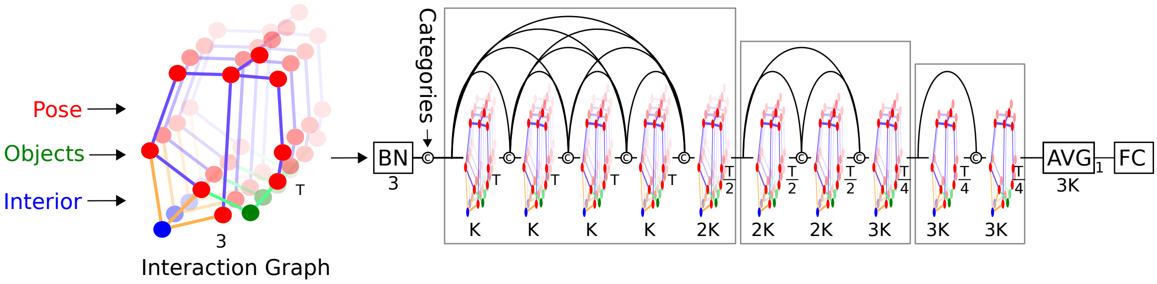


Figure 5.9: Overview of the dense spatio-temporal graph convolution network. © denotes concatenation. K denotes the output node feature length. T denotes the temporal sample length.

Our graph construction method works well with the graph convolution approach that was previously introduced. The resulting graph can still be decomposed into the neighborhood described in Section 5.3.1 by computing the distance between the center of the driver's body and object and interior nodes. This way the additional nodes become extensions of the fixed graph defined by E_{pose}. In most cases the additional edges will be part of the "further away" partition of the neighborhood (partition 2 in Equation 5.5).

5.3.3 Interaction Graph Network

Figure 5.9 shows an overview of our graph convolution-based neural network. We first apply batch normalization (BN) to the input to normalize its variance. We only apply this normalization step to the main node feature, which is the 3D position of each element, while the one-hot encoded category vector is concatenated after the normalization step. The rest of the network follows the idea of DenseNet [Hua17] by concatenating the output of all previous layers in a block as input of the current layer. Overall, the network consists of three blocks. The last spatio-temporal graph convolution of each block receives a mix of lower and higher-level features from all previous layers in the block and produces an output feature of twice the size. In addition, the temporal convolution of this layer uses a stride of two to shrink the temporal feature size by half. After three blocks this results in a feature descriptor of three times the starting feature size but with one fourth of the temporal resolution. Following the graph convolution blocks the node feature vectors of size 3K of all nodes and all time steps are aggregated using global average pooling followed by the final classification with a fully connected layer and softmax activation. Each spatio-temporal building block consists of a spatio-temporal convolution followed by batch normalization, dropout and rectified linear units as activation function. The network has two hyper-parameters, the initial feature vector length K, which also determines the size of all other layers, and the kernel size of the temporal convolution F of each spatio-temporal convolution layer. The temporal window size T of the network depends on the temporal dimension of the input sample and is not part of the parameters of the network.

5.3.4 Implementation Details

The method is implemented with the same tools and frameworks as the *Three-Stream* approach. This includes the hyper-parameter search method (see Section 5.2.3). The parameter search was conducted on the validation set of Drive&Act using the model with all features and input modalities enabled. Based on this search we train the model using stochastic gradient descent (SGD) with learning rate 0.8, Nesterov momentum of 0.9 and L2 regularization of 0.0001. We use a batch size of 64 samples and train for 50 epochs, dividing the learning rate by 10 after 20 and 40 epochs. We optimize the categorical cross entropy loss in the training process.

The main parameters of our approach were also optimized by the parameter search. Based on the results we set the edge creation threshold θ to 0.3 m. It determines the number of interaction edges created between the 3D keypoints of the driver's body, 3D interior elements, and 3D object positions. We investigate the impact of this threshold in Section 5.4.1. The size K of our neural network is set to 96 and the temporal kernal size F to 13 for all our experiments.

The configuration of the category vector, which is part of the node feature, was not part of the parameter search. Assigning a different category to each element would lead to a large vector with 36 entries (13 body keypoints, 14 interior elements, 9 objects) compared to the main node feature consisting of 3D points. Instead, we group body keypoints into four categories (head, torso, arm, hand), we leave each object in a separate category and finally assign all interior elements to a single category. This leads to a smaller one-hot encoded category vector of size 14, appended to each node feature after normalization. We assign all interior elements to a single category because they do not move. The network should be able to identify these elements based on their position. This does not work well for most keypoints of the driver's body nor 3D object positions because they move in complex patterns. A detailed category vector for these nodes should therefore help to discern them from each other.

We test the runtime performance of the system using the same hardware as the Three-Stream approach (Intel i7-9750H CPU @ 2.60GHz; GeForce GTX 1650 Max-Q) and achieve 49 fps. Compared to the runtime of our Three-Stream approach Dynamic Interaction Graphs achieve almost three times the speed while offering much higher accuracy as we will show in the following sections.

5.4 Evaluation on the Drive&Act Dataset

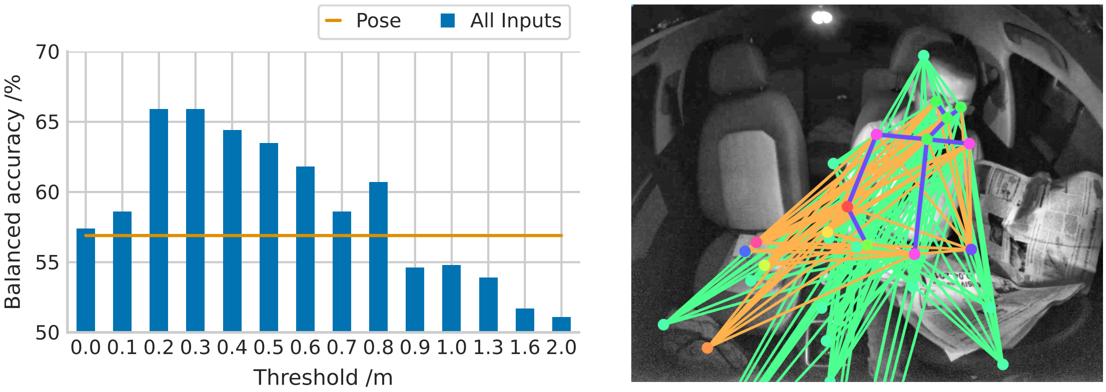
In the following we present the primary evaluation of this chapter on the Drive&Act dataset (see Section 3.5). The benchmark focusses on fine-grained classification of driver behavior. Given an action segment of three seconds or less (in case of shorter events), our goal is to assign the correct activity label. We follow standard practice and adopt the *average per-class accuracy* (in the following called *balanced accuracy*) by using the mean of the top-1 recognition rate for every category (see Section 3.5.7). Note, that the baseline of picking an activity at random is annotation level-specific and varies between 0.31 % and 16.67 %.

We evaluate our models separately for every hierarchy level: 12 scenarios/tasks (first level), 34 fine-grained activities (second level) and atomic action units with 372 possible combinations of the *{Action, Object, Location}* triplets (third level). Because the amount of triplet combinations is very high, we also report the performance for correctly classified action, object and location separately (6, 17 and 14 classes, respectively). Unless stated otherwise we use the public triangulation-based body pose annotations using the three cameras in front of the driver mounted on both A-pillars and the central mirror (see Section 3.5.4). We train our own methods using data augmentation by adding noise to all input keypoint positions with standard deviation of 0.05 m, random rotations around all axes with ±10° as well as random scaling of all input keypoints by a factor of 0.4 to 1.2. We determine the layout of our Dynamic Interaction Graph before augmentation. It therefore only affects the node features of this approach.

In the following we discuss the results of our Three-Stream method and Dynamic Interaction Graph approach. We compare our results to related methods and we perform a cross-view and cross-modal evaluation using either triangulated 3D data or 3D data created with depth images relying on our DepthFix approach.

5.4.1 Impact of the Threshold for Creating Interaction Graphs

Our method for creating Dynamic Interaction Graphs is guided by the distance of the keypoints of the driver's body to other parts of our interior state model. To create edges, we use thresholding on the distance to these parts (see Equation 5.9 θ). The threshold therefore has a large impact on the number of edges as well as the nodes included in the graph for the following driver activity recognition with the neural network based on graph convolutions. A threshold of zero means that no objects or interior elements will be included in the graph as the chance is very low that any of the driver's limbs would be placed exactly on one of those coordinates. The graph will therefore only consist of the body pose of the driver. A low threshold will connect elements in the surroundings

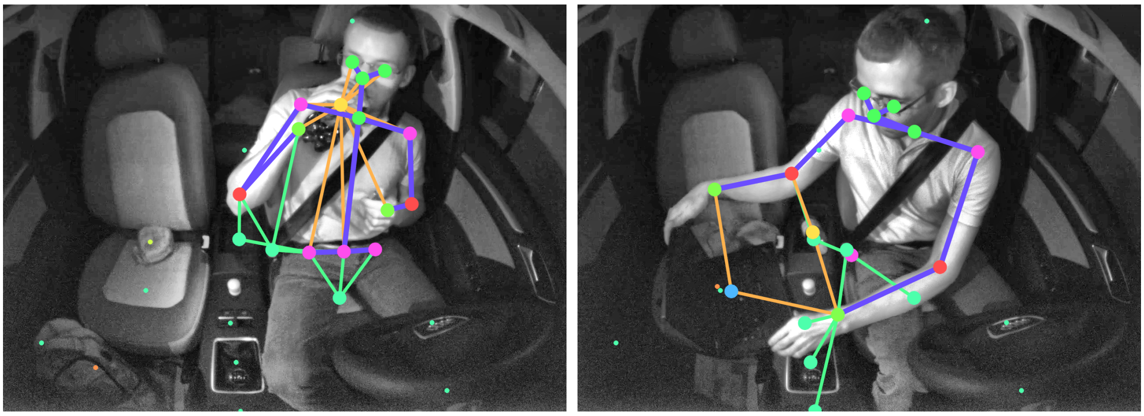


(a) Interaction Graph Threshold

(b) Threshold 2.0 m

Figure 5.10: (a) Balanced Accuracy on the validation set for different interaction thresholds using all input modalities on the recognition task of fine-grained activities. (b) Sample graph generated with a large threshold of $\theta = 2.0$ m.

if they are close to the driver and a large threshold will result in a fully connected graph between the 3D driver body pose and all other parts of the state model. We investigate the effect of this design with an ablation study on the validation set of the *fine-grained activity* annotations of Drive&Act. Figure 5.10a shows the result of this experiment using the complete interior state model as input with different interaction thresholds (blue). We compare these results to the method using just the body pose as input (orange). Note that the relevant range between 0 and 1.0 meters is sampled more densely. With a threshold of 0 the model using the complete interior state model achieves a similar result to the model using just the body pose as input. This shows that our approach works as designed because as discussed in this case no extra edges will be created so the graph will not contain objects or interior elements. A high threshold of 2 m on the other hand results in a graph that is fully connected between body pose nodes, interior and object nodes (see Figure 5.10b). This degrades the performance and highlights the importance of creating a meaningful graph layout. The method achieves the best results at a low threshold of 0.2 m to 0.3 m with an improvement of 9 percentage points compared to the baseline. This indicates that our design to create edges based on the minimum distance of all time steps is effective. This leads to graphs with few edges that may also still be interpretable by a human observer to identify problems with specific samples (see Figure 5.11). The results show a gradual decrease in performance after the first peak. This indicates that these thresholds are still suitable for a subset of the elements in the graph. An individual threshold for each element of the graph might therefore improve results further. However, testing all possible variations is not feasible and optimizing the thresholds as part of the training step of the model is not easily possible because thresholding is not differentiable. All following experiments are performed with a threshold θ of 0.3 m.



(a) Drinking

(b) Closing laptop

Figure 5.11: Reprojection of Dynamic Interaction Graphs for two different activities generated with a threshold of 0.3 m with body pose edges (blue), interior edges (orange) and object edges (green), colored circles indicate categorized locations. Small dots indicate available nodes not part of the graph.

5.4.2 Performance of Our Own Methods on Drive&Act

Our primary contribution to 3D driver body pose based activity recognition is the introduction of additional input modalities in the form of 3D interior elements and 3D object trajectories. The three annotation levels of Drive&Act pose different challenges for our methods that allow us to highlight the impact of this contribution. Table 5.2 shows the results of both our approaches for all annotation levels of the dataset and for different combinations of input modalities:

Fine-grained activities: This annotation level is the main benchmark of the Drive&Act dataset. It is challenging for methods based on just the 3D body pose of the driver because of its detailed annotations (e.g., discerning between *opening* and *closing a bottle*) resulting in small scale movements of body parts.

The performance characteristics of our Three-Stream approach on the data are overall like the results on the InCarIn activity dataset both for single streams as well as stream combinations (see Section 5.2.4). The temporal stream based on the 3D driver body pose achieves the best single stream performance while its combination with the context stream and the combination of all three streams results in the best overall performance of the method (46.6 % balanced accuracy).

Our method based on Dynamic Interaction Graphs can outperform the Three-Stream approach using just the 3D driver body pose as input (51.6 % balanced accuracy). This shows the advantage of graph convolution-based methods for activity recognition. However, unlike in our previous approach, adding just 3D interior elements decreases performance. We assume the reason is the way the information is added. While the depth distance feature (see Section 5.1) used as input of our Three-Stream approach models the distance of keypoints to the surface of interior elements, our method based on Dynamic Interaction Graphs only uses the center of each interior element as input. It therefore has no notion of the size of each element and in addition it must learn higher level features based on the training data instead of relying on the hand-crafted representation. On the other hand, adding just 3D object locations to the input improves results by a large margin as it allows to disambiguate activities with similar body pose but involving different objects. We investigate this in more detail in Figure 5.12. Adding both interior elements and objects improves results further. We assume in this case the information about the interior helps because it allows the network to correlate object positions with storage spaces like the co-driver seat or center console. Finally, adding the category description to each node feature yields the overall best result (60.4 % balanced accuracy) indicating that this descriptor helps the neural network to disambiguate different node types as we intended.

Scenarios/Task-Level: This level represents the tasks the test participants had to solve. Due to the high abstraction level, we presume that the recognition would strongly benefit from a time window longer than the current three second segments. Activities like eating and drinking are part of multiple tasks. Therefore, the shorter the classified segments the more likely it is that samples from different tasks depict the same activity. However, there can still be situational differences in the surroundings of the driver. For example, even when the driver is drinking while working on the laptop (task *working*), the laptop remains present. On the other hand, it is unlikely to be there while the driver is watching a video on the central screen (task *watching video*). Because of these challenges the overall performance is lower compared to the evaluation on *fine-grained activities* and the differences between the results are smaller and less defined. Our method based on Dynamic Interaction Graphs can only outperform the Three-Stream approach with the addition of object locations and interior elements as input modality. However, this boosts the performance by a large margin achieving the overall best result (42.3 % balanced accuracy).

Atomic Action Units: This annotation describes primitive activities that are basic building blocks of the other annotation levels. The separation of labels into triplets of *{Action, Object, Location}* allows us to investigate the impact of our input modalities in more detail because interior elements and object locations correlate directly with the respective annotation. While we expect the performance to increase in general with added inputs, the performance for object and location estimation should rise the most when adding the respective input modality.

The Three-Stream approach only uses the 3D body pose and interior elements as input. The context stream can outperform the temporal stream for action and location

Table 5.2: Results of our own methods on all three levels of the Drive&Act dataset. Both methods were tested on the public annotations using triangulation. Dynamic Interaction Graphs were additionally tested on the depth based annotations using DepthFix. Best results per section marked bold. Overall best result are underlined. Dynamic Interaction Graph Input: int: interior elements; obj: objects; cat: category descriptor. (Balanced accuracy in %)

Method	Fine-Grained	Task level	Atomic action units			
			Action	Object	Loc.	All
Random Baseline	2.9	8.3	16.7	5.9	7.1	0.3
Three-Stream LSTM (Origin: Interior Mirror; Data: Triangulated)						
Spatial	35.2	24.0	41.5	32.7	36.9	4.6
Context	40.2	29.8	49.0	40.7	53.3	6.9
Temporal	44.4	32.4	47.7	41.7	52.6	7.1
Temporal + Spatial	45.4	34.8	48.8	42.8	54.7	7.1
Temporal + Context	46.4	33.5	**51.2**	44.9	55.7	8.0
Three-Stream	**46.6**	**35.5**	50.7	**45.3**	**56.5**	**8.1**
Dynamic Interaction Graph (Origin: Interior Mirror; Data: Triangulated)						
Pose	51.6	32.7	51.6	45.6	52.4	8.9
Pose + int	49.5	31.7	48.2	44.5	55.0	7.4
Pose + obj	58.1	33.3	50.7	56.2	44.7	10.8
Pose + obj + int	59.2	40.7	58.1	56.0	50.1	13.4
Pose + obj + int + cat	**60.4**	**42.3**	**58.1**	**57.6**	**56.2**	**15.2**
Dynamic Interaction Graph (Origin: Kinect; Data: DepthFix)						
Pose + obj + int + cat	60.1	35.2	52.5	56.4	53.2	15.0

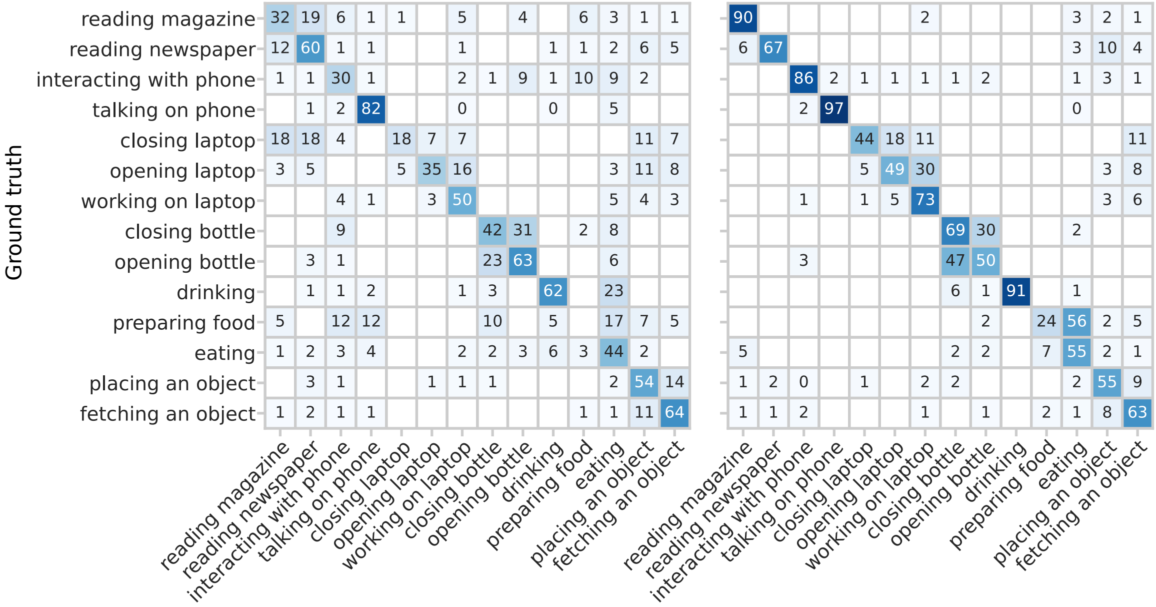
estimation while the temporal stream still achieves the best overall result. In addition, combining the temporal and context stream leads to the best performance for action estimation while the combination of all three streams results in the best result for all other annotations of this level. The Three-Stream approach can achieve the best overall result for estimating the location where an action takes place and can even outperform our newer method using Dynamic Interaction Graphs. This highlights the usefulness of our distance-based feature providing additional context for 3D body pose based activity recognition methods. Comparing all parts of the triplet the method underperforms for detecting the object involved in an activity. This is expected as the input to the method does not provide any suitable information for this detection. It can only infer the object based on correlating the provided body pose with the interior elements.

Our method based on Dynamic Interaction Graphs expands the input with additional 3D object trajectories. Similar to the other annotation levels the addition of interior elements to the input decreases the performance except for estimating the location. This indicates that the method is at least able to use this information if there is a clear correlation between the input and the annotation. Adding only 3D object positions to the input boosts the performance in general, except for detecting the location. This configuration achieves the largest performance gain for estimating the object involved in activities. This highlights the importance of this input for the task. Combining all inputs and adding the category descriptor to each node results in the overall best performance. In contrast to the Three-Stream approach our new method achieves equal performance on all parts of the annotation triplet indicating that our full interior state model, including 3D object positions, represents all annotated aspects of the *atomic action units*.

Interaction Graphs based on depth data: While our main evaluation uses the public triangulated 3D body pose data of the driver as well as the triangulated 3D object tracks we also evaluate our method using just data from the Kinect including the results of DepthFix for 3D driver body pose estimation and the 3D object tracks generated using just the depth image (see Section 3.5.6). The field of view of the Kinect is substantially smaller compared to the multi-view system. In addition, occlusions affect the Kinect data to a larger degree. Therefore, the results using the Kinect are slightly worse for all annotations compared to the results on the triangulated data. In most cases the difference is small except for the task level and the action annotation of the atomic action unit level. We suspect that this is a result of the smaller field of view as both these annotations rely on the larger context to disambiguate different activities. This experiment also shows that our DepthFix method can generate 3D body pose data from a single depth image that is comparable for activity recognition to data created with a much more complex multi-view camera setup. We investigate the cross-view performance of our Dynamic Interaction Graph approach using these camera setups in Section 5.4.4.

Overall, the results of both methods highlight the importance of a detailed 3D interior state model including 3D interior elements and 3D object trajectories to enable detection of fine-grained driver activities. Our Three-Stream method can make better use of the data about 3D interior elements compared to our method based on Dynamic Interaction Graphs. This highlights the quality of the distance feature descriptor introduced at the start of the chapter (see Equation 5.1). The performance of our method based on Dynamic Interaction Graphs demonstrates the importance of 3D object positions for fine-grained activity recognition. In addition, combining all input modalities with the category vector successfully enables the method to disambiguate different node types in the graph which allows the method to also benefit from interior elements to some degree. We could also demonstrate consistent results for both camera setups using either triangulated 3D data or 3D data inferred using depth images.

We further analyze the performance of Dynamic Interaction Graphs on the *fine-grained activity* annotation by comparing the performance of using the method with just the



	reading magazine	reading newspaper	interacting with phone	talking on phone	closing laptop	opening laptop	working on laptop	closing bottle	opening bottle	drinking	preparing food	eating	placing an object	fetching an object
reading magazine	32	19	6	1	1		5		4		6	3	1	1
reading newspaper	12	60	1	1			1			1	1	2	6	5
interacting with phone	1	1	30	1			2	1	9	1	10	9	2	
talking on phone		1	2	82			0			0		5		
closing laptop	18	18	4		18	7	7						11	7
opening laptop	3	5			5	35	16					3	11	8
working on laptop			4	1		3	50					5	4	3
closing bottle			9					42	31		2	8		
opening bottle		3	1					23	63			6		
drinking		1	1	2			1	3		62		23		
preparing food	5		12	12				10		5		17	7	5
eating	1	2	3	4			2	2	3	6	3	44	2	
placing an object		3	1			1	1	1				2	54	14
fetching an object	1	2	1	1							1	1	11	64

(a) Estimation Body Pose

	reading magazine	reading newspaper	interacting with phone	talking on phone	closing laptop	opening laptop	working on laptop	closing bottle	opening bottle	drinking	preparing food	eating	placing an object	fetching an object
reading magazine	90						2					3	2	1
reading newspaper	6	67										3	10	4
interacting with phone			86	2	1	1	1	1	2			1	3	1
talking on phone			2	97								0		
closing laptop					44	18	11							11
opening laptop					5	49	30						3	8
working on laptop			1		1	5	73						3	6
closing bottle								69	30			2		
opening bottle			3					47	50					
drinking								6	1	91		1		
preparing food									2		24	56	2	5
eating	5							2	2		7	55	2	1
placing an object	1	2	0		1		2	2				2	55	9
fetching an object	1	1	2				1		1		2	1	8	63

(b) Estimation All Inputs

Figure 5.12: Excerpt of the confusion matrices for fine-grained activity recognition using just the body pose (a) or all input modalities (b). See Appendix B for the complete matrices. For better readability values of zero are not plotted.

3D body pose of the driver to using the method with the complete interior state model. Figure 5.12 shows an excerpt of the confusion matrices of both configurations (i.e., the rows do not sum up to 100 %). To improve the readability, we only include the activities with the greatest performance gain and the activities they are most frequently confused with. The complete confusion matrices are shown in Appendix B. When comparing both results, the confusion between classes strongly decreases when using all of the inputs instead of just the body pose of the driver. Using just the body pose, activities are confused with each other if the body pose is similar (e.g., reading magazine vs. reading newspaper) or if the body pose is not distinctive (e.g., *preparing food* or *interacting with phone*). Many indistinct activities are performed in front of the torso of the driver with slightly raised hands. All these activities are confused with each other using just the body pose as input. Using the method with all inputs enabled clears up these confusions in many cases. For example, *reading a magazine* and *reading a newspaper* can then be disambiguated successfully. *Interacting with phone* is also a well detected class when including object information. It is important to note that the objects provided as input do not map one to one to a single activity in many cases. For example, there are five activities involving the laptop (only three are shown in the excerpt). With added object information the confusion between classes is more clustered for activities involving the same object. However, in some cases our interior state model is still not sufficiently detailed to resolve fine details (e.g., opening vs. closing bottle).

Table 5.3: Comparison of related methods with our results on Drive&Act. Balanced accuracy (%).

Methods	**Fine-Grained**	**Task Level**	**Atomic action units**			
			Action	Object	Loc.	All
Video						
C3D [Tra15]	43.4	-	-	-	-	-
P3D Net [Qiu17]	45.3	-	-	-	-	-
I3D Net [Car17]	63.6	31.8	56.1	56.2	51.1	12.1
CTA-Net [Wha21]	65.3	52.3	56.4	59.2	63.0	49.4
Driver Body Pose						
Two-Stream [Wan17][1]	45.4	34.8	48.8	42.8	54.7	7.1
ST-GCN [Yan18][2]	45.3	-	-	-	-	-
Three-Stream [Mar18b]	46.6	35.5	50.7	45.3	56.5	8.1
Interaction Graph [Mar20a]	60.1	42.3	58.1	57.6	56.2	15.2
Video + Driver Body Pose						
BPAI-Net [Tan21]	67.8	-	-	-	-	-

5.4.3 Comparison to Related Methods

In this section we compare our methods based on the 3D body pose of the driver and other 3D data modalities to the video-based baseline methods that were part of our initial publication of the Drive&Act dataset. In addition, we compare methods that were published til the end of 2021 using Drive&Act in their evaluation for activity recognition (see Table 5.3).

Fine-grained activities: This annotation level is used by most related methods. We also provided the the largest number of baseline methods for this annotation level including three video-based approaches (C3D, P3D Net, I3D Net). For all other annotation levels, we only provided video-based results of I3D because it was the best performing method for detecting fine-grained activities. Compared to I3D we can reduce the performance difference from 17 p.p. (percentage points) when using the Three-Stream approach to just 3.5 p.p. using Dynamic Interaction Graphs. Both our approaches outperform their respective origin methods developed for general 3D body pose-based activity recognition. The Two-Stream approach is our reimplementation of Wang et al. [Wan17] combining the temporal and spatial stream of our approach. It was the basis of our Three-Stream method which added the context stream. The ST-GCN approach [Yan18] was the basis of many related methods including our method using Dynamic Interaction Graphs. We only reused their formulation of the spatio-temporal graph convolution layer. The graph itself and the complete network structure of our method are different, improving the performance by a large margin. Following our

[1] our implementation: temporal + spatial stream

[2] reported by [Tan21]

last publication on the topic, CTA-Net further improved the results of video-based methods relying on a special convolutional network, called a glimpse sensor, to extract data from each image followed by a recurrent neural network module based on LSTM units to aggregate data temporally. BPAI-Net can achieve the overall best performance on this annotation level by combining a video-based neural network using 3D convolutions with ST-GCN for body pose-based activity recognition.

Scenarios/Task-Level: Both our methods perform better on this annotation level compared to I3D Net, our original video-based baseline approach. However, CTA-Net improves the video-based performance by a large margin, surpassing our methods.

Atomic Action Units: Both our methods were able to outperform I3D for estimating the location of the annotated triplets thanks to our explicit modeling of the position of interior elements. While the performance of our Three-Stream method suffered overall from the missing input data about objects, our method based on Dynamic Interaction Graphs alleviates this problem and outperforms I3D for this annotation level in general. However, CTA-Net further improves the performance of video-based methods, surpassing our results. We suspect that their large performance gain for estimating the whole triplet is based on training to recognize the whole triplet at once while our methods as well as the I3D baseline used separate models for each part of the triplet. Their method focused on estimating whole triplets correctly while our approaches only optimized the detection of one part of the triplet at a time.

5.4.4 Cross-View and Cross-Modal Evaluation

The goal of this thesis is the development of a modular driver activity recognition system based on 3D data from the ground up. We argued that this would greatly increase the flexibility of the system regarding sensor type and sensor position changes. However, so far, we have only shown the performance of our methods trained and tested on the same camera modalities as well as viewpoints. In the following we therefore explore the flexibility of our modular system with regards to sensor modality and viewpoint changes. To achieve this, we perform a test on Drive&Act in a cross-view setting using either the triangulation-based data without the Kinect or the depth image-based data using only the Kinect sensor as input. The triangulation-based system has a wide field of view covering both front seats and in addition its coordinate origin is the camera at the central mirror. The Kinect is mounted at the A-pillar on the co-driver side which also determines the camera coordinate system of the depth image-based 3D data. Compared to the triangulation-based system the field of view is much smaller covering mainly the driver. In addition the data is more affected by occlusions because of the single sensor setup at a position resulting in a challenging side view of the driver.

Table 5.4: Cross-view evaluation comparing different 3D augmentation methods of our Dynamic Interaction Graph approach with the performance of I3D on the validation set of *fine-grained* activities of Drive&Act. Balanced accuracy (%).

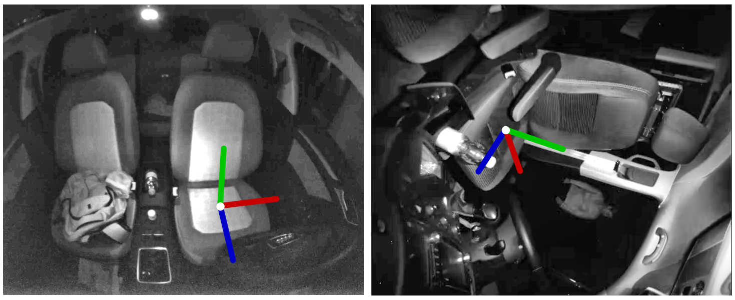
Train	**Test**							
	Video I3D		**Bodypose default**		**Bodypose ±90°**		**Bodypose seat**	
	CM	KIR	CM	KIR	CM	KIR	CM	KIR
Central Mirror (CM)	69.6	6.8	65.9	11.1	53.6	44.4	63.9	55.2
Kinect IR (KIR)	6.7	72.9	11.5	70.1	43.9	61.6	52.4	69.5

Table 5.4 shows the results compared to the performance of the video-based end-to-end model I3D. It was trained either on video data of the near-infrared camera at the central mirror or near-infrared data of the Kinect at the co-driver A-pillar. These views correspond to the coordinate systems of our 3D input data. While I3D achieves great performance on the training view, performance decreases by 90 % on average when testing on the view not used for training. Our method based on Dynamic Interaction Graphs shows similar behavior in its default configuration. However, while the performance on the training view is 5 % lower on average compared to I3D it only drops by 83 % when evaluated across views. In the following we use the performance of our default configuration tested on the training view as baseline for comparison. As our method relies on 3D data we can use 3D transformations to increase its robustness to viewpoint changes.

An approach often used by related methods is augmentation with random rotations. We test this approach by perturbing the training data by ±90° on all axes (*bodypose* ±90° in Table 5.4). With regards to the camera on the central mirror this augmentation covers all camera positions and orientations in front of the driver. This approach reduces the performance by 15 % on average when testing on the same view, but only by 35 % when evaluating across views (compared to 83 % when not perturbing the training data).

Within the interior of the car the sensor position is usually well defined both for the training set as well as for potential new camera positions. In our next approach, we therefore define a sensor independent driver seat coordinate system with its origin at the center of the seating surface and its orientation aligned with the main axes of the car (*bodypose seat* in Table 5.4). We transform the depth and triangulation-based data to the seat coordinate system for training and testing (see Figure 5.13b). As both data sources are from the same dataset and recorded in the same vehicle, this results in matching coordinate systems for both setups. However, they still differ in their field of view and quality of the data. With this approach we can achieve similar performance as our baseline system when testing on the same view and we can reduce the performance drop to 20 % when testing across views. Transforming into a common coordinate system therefore outperforms augmentation with random rotations. In addition, both augmentation approaches can outperform I3D by a large margin in a cross-view setting by making effective use of well-defined coordinate systems.

Train	**Test**	
	Cross Dataset	
	CM	KIR
D&A (CM)	80.6	46.8
InCarIn (KIR)	44.3	75.0



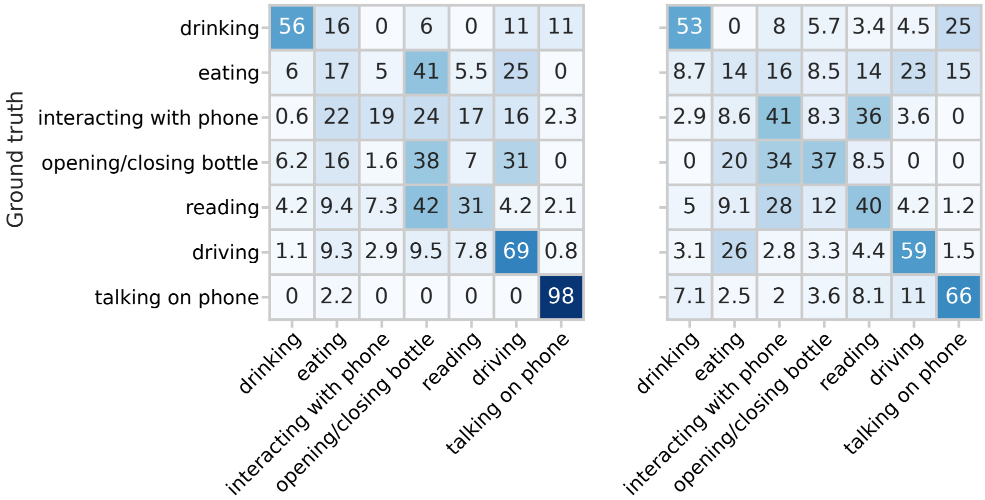
(a) Cross dataset

(b) Drive&Act (CM)

(c) InCarIn Activity (KIR)

Figure 5.13: (a) Performance of our Dynamic Interaction Graph method in a cross dataset evaluation using modified Drive&Act and InCarIn Activity datasets. Balanced accuracy (%). (b/c) Images representing the camera coordinate systems used for each dataset in the cross-dataset experiment including an annotations with the common seat coordinate system.

With our datasets we can increase the difficulty further and can perform a cross-dataset evaluation using the Drive&Act and the InCarIn activity dataset after some changes to make them more alike. The InCarIn activity dataset stays mostly the same, but we resample the data using three second segments instead of ten second segments. In addition, we determine the 3D body pose of the driver using the DepthFix method to have a comparable set of keypoints for each dataset. As a side note DepthFix is not specifically trained for this and we use the same model trained and tested on Drive&Act data. The InCarIn activities are a small subset of the *fine-grained* activity annotations of Drive&Act. We therefore create a slimmed down version of Drive&Act with six classes corresponding to the InCarIn activity annotations. To achieve this, we aggregate *reading magazin* and *reading newspaper* into one class *reading*, we combine *opening* and *closing a bottle* into one class, we rename the *sitting still* class to *driving* and we use the *interacting with phone* and *talking on phone* classes unchanged. Data of Drive&Act with other labels is not used in this experiment. With these changes we have corresponding labels for both datasets and can perform a cross-dataset evaluation. However, there are still major differences in the execution of each activity as Drive&Act was recorded in a simulator using a scenario for automated driving while the InCarIn dataset was recorded on a test track driving manually. The dimensions of both vehicles are also vastly different (Audi A3 vs. VW T5 Minibus). We still keep the cross-view setup using the triangulation-based system with its origin at the central mirror for Drive&Act and the Kinect mounted at the A-pillar for the InCarIn dataset. We combine both augmentation approaches and use the seat coordinate system in addition to applying random rotations with ±40° around all three axes. Combining both augmentation methods is helpful because the vehicles are different so the seat coordinate systems do not align perfectly (see Figure 5.13b and Figure 5.13c). A moderate amount of random rotations helps to compensate for this. Table 5.13a shows the results using just the 3D body pose data as input. We can achieve high scores for both datasets. However, in the cross-dataset evaluation performance drops by 41 % on



Ground truth	drinking	eating	interacting with phone	opening/closing bottle	reading	driving	talking on phone
drinking	56	16	0	6	0	11	11
eating	6	17	5	41	5.5	25	0
interacting with phone	0.6	22	19	24	17	16	2.3
opening/closing bottle	6.2	16	1.6	38	7	31	0
reading	4.2	9.4	7.3	42	31	4.2	2.1
driving	1.1	9.3	2.9	9.5	7.8	69	0.8
talking on phone	0	2.2	0	0	0	0	98

(a) Train: Drive&Act; Test: InCarIn

Ground truth	drinking	eating	interacting with phone	opening/closing bottle	reading	driving	talking on phone
drinking	53	0	8	5.7	3.4	4.5	25
eating	8.7	14	16	8.5	14	23	15
interacting with phone	2.9	8.6	41	8.3	36	3.6	0
opening/closing bottle	0	20	34	37	8.5	0	0
reading	5	9.1	28	12	40	4.2	1.2
driving	3.1	26	2.8	3.3	4.4	59	1.5
talking on phone	7.1	2.5	2	3.6	8.1	11	66

(b) Train: InCarIn; Test: Drive&Act

Figure 5.14: Cross-Dataset performance of our Dynamic Interaction Graph approach.

average. Figure 5.14 shows the confusion matrices resulting from this experiment. The performance of distinctive activities (e.g., *talking on phone*) stays high across datasets while activities performed differently do not work well (e.g., *eating*). Given the major differences between the vehicles and the recorded activities we still think this is a good result that demonstrates the generalization capabilities of our method.

Until now we just discussed our experiments with regards to view-point changes. However, these experiments are also cross-modal evaluations, which means the input modality of the system changes. For end-to-end video-based methods this usually means, for example, training on color data and testing on near-infrared data. Reiß et al. [Rei20a] perform such an experiment using the data from the Kinect of the Drive&Act dataset. All sensors of the Kinect are in close proximity to each other, so this is mostly a cross-modal and not a cross-view experiment. They train on color data and test on near-infrared images (see Table 5.5). Using the baseline I3D training approach the performance drops by 77 % in the cross-modal setting. Therefore, they introduced a video translation method that converts color data to near-infrared-like data for training. This doubles their cross-modal performance but still results in a 57 % decrease compared to the baseline. With regards to our algorithms, the input to our activity recognition methods is based on the 3D driver body pose in all cases. However, the whole idea of our approach is modularity of the overall system. For the system as a whole, our cross-view experiments are also cross-modal experiments as we switch from a multi-view system to a depth camera to generate 3D data. As already shown, this only degrades the results by 20 % even with an additional viewpoint change when evaluating on the same dataset (*bodypose seat* in Table 5.4). This is a far better result compared to the cross-modal performance of I3D even with video translation. The performance of our method only drops by 41 % when switching to a different dataset while changing the viewpoint and modality (see Table 5.13a).

Table 5.5: Performance of I3D in a cross modal setup training on the Kinect color data and testing on the Kinect NIR data of Drive&Act [Rei20a]. $T()$ indicates video image domain translation before training. Balanced accuracy (%).

Cross-Modal Video I3D		
Train	**Test**	**Bal. Acc.**
Kinect Color	Kinect Color	67.8
Kinect Color	Kinect IR	15.6
T(Kinect Color)	Kinect IR	29.3

However, this result is not directly comparable to the cross-modal performance of I3D as it uses a modified dataset with fewer classes. In addition, all our methods for creating the 3D body pose of the driver rely on OpenPose applied to near-infrared images. However, it was trained on color data and we demonstrated its cross-modal performance using the Drive&Act body pose benchmark in Section 4.4.1. Therefore our use of OpenPose is cross-modal in itself. We would therefore claim that our approach would achieve similar performance when applied to a multi-view camera system using color cameras or a depth camera providing a color image in addition to the depth image. However, we can not test these claims on the available data.

To summarize the findings of this section, we were able to demonstrate the cross-view and cross-modal capabilities of our modular overall system by changing between near-infrared-based triangulation and depth image-based data using the Drive&Act dataset. For both scenarios we compared it with I3D, an end-to-end video-based method for activity recognition. While the performance of I3D dropped severely in a cross-view evaluation, our approach retained most of its performance by using coordinate transformations for augmentation with random rotations or normalization of the data across views with a common coordinate system. For cross-modal evaluation we compared our modular system with the baseline I3D approach as well as with a method using video domain translation for training I3D. While this increased the cross-modal capabilities of I3D substantially, we achieved even better results with the same setup used in the cross-view experiments. In addition, we also demonstrated the ability of our method to generalize across datasets recorded in different vehicles in addition to changing the view point and the input video data modality.

5.5 Summary

In this chapter we presented our contributions for 3D driver body pose-based activity recognition using our collected datasets and body pose estimation methods. Our primary focus was the introduction of additional input modalities. To this end we extended the 3D body pose-based input of related methods to a 3D interior state model including additional interior elements, like controls and seats, and locations of objects used for various activities, like smartphones or bottles. We investigated several ways to integrate these modalities into our activity recognition systems. We followed one central paradigm and assumed that the distance in 3D space of keypoints of the driver's body to other elements of the state model is an important indicator of their relevance for activity recognition. With this framework we introduced three methods.

At first, we tested this paradigm using an analytical method to detect interactions between the driver and the surrounding interior, like hands on wheel detection. The method relied solely on the distance between the body pose and interior elements in 3D space using thresholding to determine if the driver interacts with an element. We demonstrated reliable performance for this task using this method.

In the following we used the same distance estimation to integrate interior elements for driver activity recognition using recurrent neural networks in three streams with late fusion. We demonstrated performance gains using this approach on both our datasets. However, on Drive&Act we determined that the performance for some activities suffered because of missing data about objects the driver interacts with.

Our final method using Dynamic Interaction Graphs therefore expanded the interior state model with additional 3D object trajectories. We used all three input modalities to generate a spatio-temporal graph using a neural network based on graph convolutions to infer activities. Our primary contribution with this method was the approach to creating the graph. Following our initial principle it was guided by the distance between the driver and surrounding elements to determine the nodes and edges of the graph. We demonstrated the usefulness of this graph creation method on Drive&Act. In addition, we showed that the use of 3D object positions as input helped to disambiguate many activities of the Drive&Act dataset with similar or minor body movements. Both measures improved the performance of this approach by a large margin compared to our previous method.

Finally, we showed the advantage of our overall modular architecture in a cross-view and cross-modal test demonstrating high performance even when switching between data from a multi-view system and data from just one depth camera. We also demonstrated the performance of our method across datasets recorded in a simulator or in a different vehicle on a test track.

6 Conclusion

In this thesis we researched how to detect the activities of drivers in automated cars using a modular activity recognition system based on 3D data. This system consists of two stages. The first stage creates a 3D interior state model from camera data including the 3D body pose of the driver, the location of elements of the interior as well as the position of objects involved in certain activities. The second stage uses this representation to detect activities. We based this design on two hypotheses: First, a rich 3D interior state model including other elements in addition to the 3D driver body pose is important to discern fine-grained activities. Second, the interior state model, created by the first stage, is a sensor modality and sensor location independent representation allowing the second stage to successfully detect activities even with major changes to the input of the overall system. To verify these assumptions, we contributed to the research field in three areas:

The foundation of all our efforts were annotated datasets. Based on our extensive literature review we could show that there were no suitable public data sources. We therefore collected multiple datasets for different sub-tasks leading to our final dataset, published under the name Drive&Act. It included a large-scale hierarchical activity recognition benchmark with multiple 3D input modalities for the task, like the 3D body pose of the driver, the location of interior elements, like the steering wheel, as well as 3D trajectories of objects, like smartphones. In addition, the dataset included a public benchmark for 3D driver body pose estimation with challenging scenes of different activities involving occluding objects.

The main feature of the 3D interior state model was the body pose of the driver. Here we contributed methods for real-time 3D driver body pose estimation based on depth images. The primary challenge using depth images for this task was occlusion of body parts. A second challenge was the lack of public automotive datasets to train these methods. Our first approach used random decision forests for body part labeling based on a segmentation of the driver in the depth image. We dealt with the data scarcity by using simulated depth images for training. While we achieved satisfactory results with this approach in general, the necessary segmentation as well as the lack of occlusion handling limited its usefulness as input for our activity recognition methods. Our second approach focused on handling these cases with a regression-based method using deep learning. Unfortunately, this also increased the demand for large-scale datasets. To address this issue, we relied on a novel split of 2D body pose estimation followed by separate 3D keypoint regression guided by the depth image. This enabled us to rely on advances in 2D body pose estimation using large-scale datasets from other domains. We demonstrated the robustness of this method to partial occlusion on the body pose benchmark of Drive&Act.

While the primary input of many related activity recognition methods was just the 3D human body pose, we researched how to expand the input to a complex 3D state model including elements of the surrounding vehicle interior as well as positions of objects relevant for certain activities. We followed one central paradigm and assumed that the distance in 3D space of keypoints of the driver's body to other elements in the state model is an important indicator of their relevance for the performed activity. Based on this hypothesis we developed different methods with increasing complexity of the interior state model. Our first approach for detecting interactions between the driver and interior elements (e.g., hands on wheel detection) relies on thresholding of the distance between the hands of the driver and the respective interior elements. With this approach we demonstrated the usefulness of the general concept. We then expanded the idea combining the distance of keypoints of the driver's body to interior elements with recurrent neural networks to detect activities. While we successfully demonstrated the usefulness of interior elements for this task, we also showed that our method was unable to discern between activities with similar body poses but involving different objects. We therefore expanded the input with additional 3D object trajectories casting all parts of the interior state model into a spatio-temporal graph. To generate this graph, we relied on the distance of keypoints of the driver's body to other parts of the state model to determine which nodes to include in the graph and what edges to create. We analyzed this graph using a neural network based on graph convolutions. We showed the advantage of our graph creation method in selecting relevant interior elements and objects and the usefulness of object location data to discern activities represented by similar body poses of the driver. Consequently, we proved our initial hypothesis that additional input modalities improve the detection of fine-grained activities based on 3D data and we quantified their impact.

We also investigated the overall performance of our modular system regarding sensor modality and viewpoint changes. We demonstrated the capability to switch between creating the interior state model based on a multi-view camera system to creating it using data from a single depth sensor. We showed that our activity recognition approach can be trained on one of these representations and evaluated on the other with just a moderate performance drop. In addition, the system was able to generalize across different datasets recorded in different vehicles and in vastly different conditions switching between data recorded in a simulator for automated driving and data recorded on a test track driving manually. This proved our second hypothesis that the 3D interior state model resulting from our first stage of algorithms is sensor independent to a large degree.

6.1 Future Work and Open Research Questions

While we achieved great progress for driver activity recognition in automated cars, there are still open research questions that could lead to further improvements.

Regarding datasets we made significant progress in increasing the size of public datasets both for driver body pose estimation and driver activity recognition. However, our final dataset was recorded in a driving simulator. While we tested our approach on the InCarIn dataset recorded on a test track, this dataset was smaller than Drive&Act. To determine the real-world performance of our approach the next step would be to collect data in real traffic, preferably without a strict data collection protocol in order to record more natural driver behavior.

Our 3D body pose estimation methods relied on depth data or multi-view systems. While we showed that this works well, these sensor setups are less flexible and more expensive than monocular camera systems. However, there is great progress estimating depth data, 3D body pose or 3D object bounding boxes from monocular data alone. It would be interesting to see if these methods are good enough to generate our 3D interior state model while keeping the flexibility of the activity recognition methods leveraging this representation. While we showed that a rich interior state model improves activity recognition results, adding more detectors increases the necessary computing power. An interesting research direction would be multi-task learning to produce the interior state model using just one neural network with multiple heads trained on various datasets.

The input of our activity recognition system currently does not include the most common features for driver monitoring based on facial analysis like head pose estimation or eye tracking and related features like the visual focus of attention. It would be interesting to test if these inputs could further improve the results. In addition, our activity recognition systems focus on classifying short sections of data while in a real-world application the data stream is continuous. We can apply our methods to continuous data streams by generating segments, but our method never learned to deal with transitions between activities and it has no notion of the start and the end of an activity. Methods for activity detection handle these challenges and can naturally process longer sequences including transitions. Our methods also perform closed set recognition. They do not have a notion of unknown activities which is necessary in most real-world applications as not all activities performed by the driver are part of the datasets. Our methods are robust to this to some degree as the *sitting still* label of Drive&Act acts in part as a catch all class for anything that is not a labeled activity, but it would be better to adapt the recognition method for open set classification from the ground up. This could also be extended to anomaly detection, i.e. identifying situations that are outside of the expected behavior of the driver.

6.2 Applications to Other Fields

The methods presented in this thesis produce a rich representation of the surroundings including the location of regions of interest of the surrounding environment, the location of objects and the 3D body pose of the driver. From these we infer activities. There are other domains where such a system can be useful or where the representation of the surroundings can be a good starting point for other tasks instead of activity recognition.

Smart homes and IoT (Internet of Things) devices are a promising future application for our methods. Currently these systems are controlled by a combination of voice commands, using smart speakers, and various other sensors like motion detectors. While using camera systems increases privacy concerns, our methods could run locally and only transmit anonymized 3D data to the cloud if necessary. This would enable smart speakers to react to the user in a much more direct way based on the precise location of people in the room and what they are doing. We assume such systems will be necessary with an increasing number of controllable devices because controlling them manually gets increasingly unfeasible and automated control based on primitive sensors might be too unreliable. While such a system can increase the general level of comfort for anyone it could also help with elder care, for example, by detecting falls or reminding people that they forgot to take their prescribed medication. There are also research projects that investigate the use of robots for elder care, which would likely require similar detection capabilities.

Our approaches could also be useful to track tasks in manufacturing and to support assembly workers. In this case our system could, for example, detect if the worker picks objects from the right box. It could also detect common manufacturing steps like assembling an object, picking it up or placing it somewhere. All this information can be used to detect errors but also to prevent them, either by guiding the worker or by preventing the continuation to the next assembly step if an error was detected. If both humans and robots are involved in a manufacturing task this gets even more important for human robot interaction because it enables the robot to react to gestures of the worker or to detect if a manufacturing step requires its support.

Bibliography

[Abo18] Abouelnaga, Yehya; Eraqi, Hesham M. and Moustafa, Mohamed N.: “Real-Time Distracted Driver Posture Classification”. In: Machine Learning for Intelligent Transportation Systems Workshop in the Conference on Neural Information Processing Systems (NeuroIPS). 2018 (cit. on pp. 35–37, 39, 40).

[Alo20] Alotaibi, Munif and Alotaibi, Bandar: “Distracted Driver Classification Using Deep Learning”. In: *Signal, Image and Video Processing* 14.3 (Apr. 1, 2020), pp. 617–624. issn: 1863-1711 (cit. on p. 39).

[And14] Andriluka, Mykhaylo; Pishchulin, Leonid; Gehler, Peter and Schiele, Bernt: “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2014, pp. 3686–3693 (cit. on pp. 14, 16, 25).

[Are19] Arefin, Md Rifat; Makhmudkhujaev, Farkhod; Chae, Oksam and Kim, Jaemyun: “Aggregating CNN and HOG Features for Real-Time Distracted Driver Detection”. In: *International Conference on Consumer Electronics (ICCE).* Jan. 2019, pp. 1–3 (cit. on p. 39).

[Baa13] Baak, Andreas; Müller, Meinard; Bharaj, Gaurav; Seidel, Hans-Peter and Theobalt, Christian: “A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera”. In: *Consumer Depth Cameras for Computer Vision: Research Topics and Applications.* Ed. by Fossati, Andrea; Gall, Juergen; Grabner, Helmut; Ren, Xiaofeng and Konolige, Kurt. Advances in Computer Vision and Pattern Recognition. London: Springer, 2013, pp. 71–98. isbn: 978-1-4471-4640-7 (cit. on p. 19).

[Bah18] Baheti, Bhakti; Gajre, Suhas and Talbar, Sanjay: “Detection of Distracted Driver Using Convolutional Neural Network”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018, pp. 1032–1038 (cit. on p. 39).

[Bah20] Baheti, Bhakti; Talbar, Sanjay and Gajre, Suhas: “Towards Computationally Efficient and Realtime Distracted Driver Detection With MobileVGG Network”. In: *IEEE Transactions on Intelligent Vehicles* 5.4 (Dec. 2020), pp. 565–574. issn: 2379-8904 (cit. on p. 39).

[Ban18] Banks, Victoria A.; Eriksson, Alexander; O'Donoghue, Jim and Stanton, Neville A.: "Is Partially Automated Driving a Bad Idea? Observations from an on-Road Study". In: *Applied Ergonomics* 68 (Apr. 1, 2018), pp. 138–145. issn: 0003-6870 (cit. on pp. 2, 3).

[Baz20] Bazarevsky, Valentin; Grishchenko, Ivan; Raveendran, Karthik; Zhu, Tyler; Zhang, Fan and Grundmann, Matthias: "BlazePose: On-device Real-time Body Pose Tracking". June 17, 2020. arXiv: 2006.10204 [cs] (cit. on p. 17).

[Beh18a] Behera, Ardhendu and Keidel, Alex: "Latent Body-Pose Guided DenseNet for Recognizing Driver's Fine-grained Secondary Activities". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* Sept. 1, 2018 (cit. on pp. 26, 41).

[Beh18b] Behera, Ardhendu; Keidel, Alex and Debnath, Bappaditya: "Context-Driven Multi-stream LSTM (M-LSTM) for Recognizing Fine-Grained Activity of Drivers". In: *Conference Proceedings from 40th German Conference on Pattern Recognition (GCPR) 9th to 12th October 2018 Stuttgart Germany.* Aug. 27, 2018 (cit. on pp. 26, 42).

[Beh20] Behera, A.; Wharton, Z.; Keidel, A. and Debnath, B.: "Deep CNN, Body Pose and Body-Object Interaction Features for Drivers' Activity Monitoring". In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–8. issn: 1558-0016 (cit. on pp. 26, 41).

[Bel14] Belagiannis, Vasileios; Amin, Sikandar; Andriluka, Mykhaylo; Schiele, Bernt; Navab, Nassir and Ilic, Slobodan: "3D Pictorial Structures for Multiple Human Pose Estimation". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014, pp. 1669–1676 (cit. on pp. 15, 25).

[Ber21] Bera, Asish; Wharton, Zachary; Liu, Yonghuai; Bessis, Nik and Behera, Ardhendu: "Attend and Guide (AG-Net): A Keypoints-Driven Attention-Based Deep Network for Image Recognition". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 3691–3704. issn: 1941-0042 (cit. on p. 39).

[Bil19] Billah, Tashrif; Rahman, S. M. Mahbubur; Ahmad, M. Omair and Swamy, M. N. S.: "Recognizing Distractions for Assistive Driving by Tracking Body Parts". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.4 (Apr. 2019), pp. 1048–1062. issn: 1558-2205 (cit. on pp. 36, 37).

[Bor18] Borghi, Guido; Frigieri, Elia; Vezzani, Roberto and Cucchiara, Rita: "Hands on the Wheel: A Dataset for Driver Hand Detection and Tracking". In: *International Conference on Automatic Face Gesture Recognition (FG 2018).* May 2018, pp. 564–570 (cit. on p. 38).

[Bor20] Borges, João; Queirós, Sandro; Oliveira, Bruno; Torres, Helena; Rodrigues, Nelson; Coelho, Victor; Pallauf, Johannes; Brito, José Henrique; Mendes, José and Fonseca, Jaime C.: "A System for the Generation of In-Car Human Body Pose Datasets". In: *Machine Vision and Applications* 32.1 (Oct. 8, 2020), p. 4. issn: 1432-1769 (cit. on pp. 23–27, 69).

[Bra00] Bradski, G.: "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000) (cit. on p. 97).

[Buy14] Buys, Koen; Cagniart, Cedric; Baksheev, Anatoly; De Laet, Tinne; De Schutter, Joris and Pantofaru, Caroline: "An Adaptable System for RGB-D Based Human Body Detection and Pose Estimation". In: *Journal of Visual Communication and Image Representation.* Visual Understanding and Applications with RGB-D Cameras 25.1 (Jan. 1, 2014), pp. 39–52. issn: 1047-3203 (cit. on pp. 18, 19).

[Cae19] Caetano, Carlos; Sena, Jessica; Brémond, François; Dos Santos, Jefersson A. and Schwartz, William Robson: "SkeleMotion: A New Representation of Skeleton Joint Sequences Based on Motion Information for 3D Action Recognition". In: *International Conference on Advanced Video and Signal Based Surveillance (AVSS).* Sept. 2019, pp. 1–8 (cit. on p. 32).

[Cañ21] Cañas, Paola; Ortega, Juan Diego; Nieto, Marcos and Otaegui, Oihana: "Detection of Distraction-Related Actions on DMD: An Image and a Video-Based Approach Comparison." In: *VISIGRAPP (5: VISAPP).* 2021, pp. 458–465 (cit. on pp. 36, 39, 40).

[Cao17] Cao, Z.; Simon, T.; Wei, S. and Sheikh, Y.: "Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* July 2017, pp. 1302–1310 (cit. on pp. 18, 20, 27).

[Cao18] Cao, Zhe; Hidalgo, Gines; Simon, Tomas; Wei, Shih-En and Sheikh, Yaser: "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". 2018. arXiv: 1812.08008 [cs] (cit. on pp. 18, 20, 69, 97).

[Car17] Carreira, Joao and Zisserman, Andrew: "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2017, pp. 6299–6308 (cit. on pp. 29, 30, 37, 134).

[Çet21] Çetinkaya, Mert and Acarman, Tankut: “Driver Activity Recognition Using Deep Learning and Human Pose Estimation”. In: *International Conference on INnovations in Intelligent SysTems and Applications (INISTA).* Aug. 2021, pp. 1–5 (cit. on p. 42).

[Che18] Chen, Yilun; Wang, Zhicheng; Peng, Yuxiang; Zhang, Zhiqiang; Yu, Gang and Sun, Jian: “Cascaded Pyramid Network for Multi-Person Pose Estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 7103–7112 (cit. on p. 16).

[Che20a] Chen, Ju-Chin; Lee, Chien-Yi; Huang, Peng-Yu and Lin, Cheng-Rong: “Driver Behavior Analysis via Two-Stream Deep Convolutional Neural Network”. In: *Applied Sciences* 10.6 (6 Jan. 2020), p. 1908. issn: 2076-3417 (cit. on p. 40).

[Che20b] Cheng, Ke; Zhang, Yifan; Cao, Congqi; Shi, Lei; Cheng, Jian and Lu, Hanqing: “Decoupling GCN with DropGraph Module for Skeleton-Based Action Recognition”. In: *Computer Vision – ECCV 2020.* Ed. by Vedaldi, Andrea; Bischof, Horst; Brox, Thomas and Frahm, Jan-Michael. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 536–553. isbn: 978-3-030-58586-0 (cit. on p. 34).

[Chu19] Chun, Sehyun; Hamidi Ghalehjegh, Nima; Choi, Joseph; Schwarz, Chris; Gaspar, John; McGehee, Daniel and Baek, Stephen: “NADS-Net: A Nimble Architecture for Driver and Seat Belt Detection via Convolutional Neural Networks”. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2019 (cit. on pp. 23–25, 27).

[Com15] Commission, European; Mobility, Directorate-General for and Transport: Study on Good Practices for Reducing Road Safety Risks Caused by Road User Distractions : Final Report. Publications Office, 2015 (cit. on p. 2).

[Com18] Community, Blender Online: Blender - a 3D Modelling and Rendering Package. manual. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018 (cit. on p. 51).

[Cro17] Cronje, Jaco and Engelbrecht, Andries P.: “Training Convolutional Neural Networks with Class Based Data Augmentation for Detecting Distracted Drivers”. In: *Proceedings of the 9th International Conference on Computer and Automation Engineering.* ICCAE ’17. New York, NY, USA: Association for Computing Machinery, Feb. 18, 2017, pp. 126–130. isbn: 978-1-4503-4809-6 (cit. on p. 25).

[Dam12] Damböck, Daniel; Farid, Mehdi; Tönert, Lars and Bengler, Klaus: "Übernahmezeiten Beim Hochautomatisierten Fahren". In: *Tagung Fahrerassistenz. München* 15 (2012), p. 16 (cit. on pp. 4, 5).

[Das15] Das, Nikhil; Ohn-Bar, Eshed and Trivedi, Mohan M.: "On Performance Evaluation of Driver Hand Detection Algorithms: Challenges, Dataset, and Metrics". In: *International Conference on Intelligent Transportation Systems (ITSC).* Sept. 2015, pp. 2953–2958 (cit. on pp. 23, 25).

[DeF20] DeFazio, Peter A.: H.R.2 - 116th Congress (2019-2020): Moving Forward Act. July 20, 2020. url: http://www.congress.gov/ (visited on 07/17/2022) (cit. on p. 7).

[Dem09] Demirdjian, David and Varri, Chenna: "Driver Pose Estimation with 3D Time-of-Flight Sensor". In: *Workshop on Computational Intelligence in Vehicles and Vehicular Systems.* 2009, pp. 16–22 (cit. on p. 26).

[Dia20] Dias Da Cruz, Steve; Wasenmüller, Oliver; Beise, Hans-Peter; Stifter, Thomas and Stricker, Didier: "SVIRO: Synthetic Vehicle Interior Rear Seat Occupancy Dataset and Benchmark". In: *Winter Conference on Applications of Computer Vision (WACV).* 2020 (cit. on pp. 23–25).

[Die22a] Diederichs, Frederik; Muthumani, Arun; Feierle, Alexander; Galle, Melanie; Mathis, Lesley-Ann; Bopp-Bertenbreiter, Valeria; Widlroither, Harald and Bengler, Klaus: "Improving Driver Performance and Experience in Assisted and Automated Driving With Visual Cues in the Steering Wheel". In: *IEEE Transactions on Intelligent Transportation Systems* 23.5 (May 2022), pp. 4843–4852. issn: 1558-0016 (cit. on p. 3).

[Die22b] Diederichs, Frederik; Wannemacher, Christoph; Faller, Fabian; Mikolajewski, Martin et al.: "Artificial Intelligence for Adaptive, Responsive, and Level-Compliant Interaction in the Vehicle of the Future (KARLI)". In: *HCI International 2022 Posters.* Ed. by Stephanidis, Constantine; Antona, Margherita and Ntoa, Stavroula. Communications in Computer and Information Science. Cham: Springer International Publishing, 2022, pp. 164–171. isbn: 978-3-031-06394-7 (cit. on p. 3).

[DiF15] DiFilippo, Nicholas M. and Jouaneh, Musa K.: "Characterization of Different Microsoft Kinect Sensor Models". In: *IEEE Sensors Journal* 15.8 (Aug. 2015), pp. 4554–4564. issn: 1558-1748 (cit. on p. 47).

[Din15] Ding, Meng and Fan, Guoliang: "Articulated Gaussian Kernel Correlation for Human Pose Estimation". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015, pp. 57–64 (cit. on p. 19).

[Din16] Dingus, Thomas A.; Guo, Feng; Lee, Suzie; Antin, Jonathan F.; Perez, Miguel; Buchanan-King, Mindy and Hankey, Jonathan: “Driver Crash Risk Factors and Prevalence Evaluation Using Naturalistic Driving Data”. In: *Proceedings of the National Academy of Sciences* 113.10 (Mar. 8, 2016), pp. 2636–2641 (cit. on p. 2).

[Don15] Donahue, Jeffrey; Anne Hendricks, Lisa; Guadarrama, Sergio; Rohrbach, Marcus; Venugopalan, Subhashini; Saenko, Kate and Darrell, Trevor: “Long-Term Recurrent Convolutional Networks for Visual Recognition and Description”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 2625–2634 (cit. on p. 30).

[Don19] Dong, Junting; Jiang, Wen; Huang, Qixing; Bao, Hujun and Zhou, Xiaowei: “Fast and Robust Multi-Person 3D Pose Estimation From Multiple Views”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 7792–7801 (cit. on p. 21).

[Du15a] Du, Yong; Fu, Yun and Wang, Liang: “Skeleton Based Action Recognition with Convolutional Neural Network”. In: *Asian Conference on Pattern Recognition (ACPR).* 2015, pp. 579–583 (cit. on p. 32).

[Du15b] Du, Yong; Wang, Wei and Wang, Liang: “Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 1110–1118 (cit. on p. 32).

[E-S19] E-Survey of Road Users’ Attitudes: “ESRA Report Distraction (Mobile Phone Use)”. June 18, 2019 (cit. on p. 2).

[Era19] Eraqi, Hesham M.; Abouelnaga, Yehya; Saad, Mohamed H. and Moustafa, Mohamed N.: “Driver Distraction Identification with an Ensemble of Convolutional Neural Networks”. In: *Journal of Advanced Transportation* 2019 (2019), e4125865. issn: 0197-6729 (cit. on pp. 36, 39).

[Eri17] Eriksson, Alexander and Stanton, Neville A.: “Takeover Time in Highly Automated Vehicles: Noncritical Transitions to and From Manual Control”. In: *Human Factors* 59.4 (June 1, 2017), pp. 689–705. issn: 0018-7208 (cit. on p. 2).

[Ese22] Esenturk, Emre; Wallace, Albert G.; Khastgir, Siddartha and Jennings, Paul: “Identification of Traffic Accident Patterns via Cluster Analysis and Test Scenario Development for Autonomous Vehicles”. In: *IEEE Access* 10 (2022), pp. 6660–6675. issn: 2169-3536 (cit. on p. 2).

[Eur19] European Union: EUR-Lex - 32019R2144 - EN - EUR-Lex. 2019. url: https://eur-lex.europa.eu/eli/reg/2019/2144/oj (visited on 07/17/2022) (cit. on p. 7).

[Ezz21] Ezzouhri, Amal; Charouh, Zakaria; Ghogho, Mounir and Guennoun, Zouhair: "Robust Deep Learning-Based Driver Distraction Detection and Classification". In: *IEEE Access* 9 (2021), pp. 168080–168092. issn: 2169-3536 (cit. on p. 39).

[Fan17] Fang, Hao-Shu; Xie, Shuqin; Tai, Yu-Wing and Lu, Cewu: "RMPE: Regional Multi-Person Pose Estimation". In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2334–2343 (cit. on p. 17).

[Fel21] Feld, Hartmut; Mirbach, Bruno; Katrolia, Jigyasa; Selim, Mohamed; Wasenmüller, Oliver and Stricker, Didier: "DFKI Cabin Simulator: A Test Platform for Visual In-Cabin Monitoring Functions". In: *Commercial Vehicle Technology 2020/2021*. Ed. by Berns, Karsten; Dressler, Klaus; Kalmar, Ralf; Stephan, Nicole; Teutsch, Roman and Thul, Martin. Proceedings. Wiesbaden: Springer Fachmedien, 2021, pp. 417–430. isbn: 978-3-658-29717-6 (cit. on p. 24).

[Gan10] Ganapathi, Varun; Plagemann, Christian; Koller, Daphne and Thrun, Sebastian: "Real Time Motion Capture Using a Single Time-of-Flight Camera". In: *Computer Society Conference on Computer Vision and Pattern Recognition*. June 2010, pp. 755–762 (cit. on p. 15).

[Gan12] Ganapathi, Varun; Plagemann, Christian; Koller, Daphne and Thrun, Sebastian: "Real-Time Human Pose Tracking from Range Data". In: *Computer Vision – ECCV 2012*. Ed. by Fitzgibbon, Andrew; Lazebnik, Svetlana; Perona, Pietro; Sato, Yoichi and Schmid, Cordelia. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 738–751. isbn: 978-3-642-33783-3 (cit. on pp. 15, 19).

[Gao12] Gao, Fuchang and Han, Lixing: "Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters". In: *Computational Optimization and Applications* 51.1 (2012), pp. 259–277 (cit. on p. 96).

[Gir11] Girshick, Ross; Shotton, Jamie; Kohli, Pushmeet; Criminisi, Antonio and Fitzgibbon, Andrew: "Efficient Regression of General-Activity Human Poses from Depth Images". In: *International Conference on Computer Vision (ICCV)*. Nov. 2011, pp. 415–422 (cit. on p. 19).

[Gol13] Gold, Christian; Damböck, Daniel; Lorenz, Lutz and Bengler, Klaus: ""Take over!" How Long Does It Take to Get the Driver Back into the Loop?" In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 57.1 (2013), pp. 1938–1942 (cit. on pp. 4, 5).

[Gor05] GORDON, Craig P.: “Driver Distraction: An Initial Examination of the ’attention Diverted by’ Contributory Factor Codes from Crash Reports and Focus Group Research on Perceived Risks”. In: 2005 (cit. on p. 2).

[Gre18] GREENLEE, Eric T.; DELUCIA, Patricia R. and NEWTON, David C.: “Driver Vigilance in Automated Vehicles: Hazard Detection Failures Are a Matter of Time”. In: *Human Factors* 60.4 (June 1, 2018), pp. 465–476. ISSN: 0018-7208 (cit. on p. 2).

[Gu22] GU, Yanlei; SONG, Yejin; GONCHARENKO, Igor and KAMIJO, Shunsuke: “Driver Hand Activity Recognition Using NIR Camera and Deep Neural Network”. In: *Global Conference on Life Sciences and Technologies (LifeTech).* Mar. 2022, pp. 299–300 (cit. on p. 38).

[Gue21] GUESDON, Romain; CRISPIM-JUNIOR, Carlos and TOUGNE, Laure: “DriPE: A Dataset for Human Pose Estimation in Real-World Driving Settings”. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021, pp. 2865–2874 (cit. on pp. 23–25, 27).

[Gup15] GUPTA, Rajeev; P, Mangalraj; AGRAWAL, Anupam and KUMAR, Anil: “Posture Recognition for Safe Driving”. In: *International Conference on Image Information Processing (ICIIP).* Dec. 2015, pp. 141–146 (cit. on p. 39).

[Haq16] HAQUE, Albert; PENG, Boya; LUO, Zelun; ALAHI, Alexandre; YEUNG, Serena and FEI-FEI, Li: “Towards Viewpoint Invariant 3D Human Pose Estimation”. In: *Computer Vision – ECCV 2016.* Ed. by LEIBE, Bastian; MATAS, Jiri; SEBE, Nicu and WELLING, Max. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 160–177. ISBN: 978-3-319-46448-0 (cit. on pp. 15, 20, 25).

[Har04] HARTLEY, Richard and ZISSERMAN, Andrew: Multiple View Geometry in Computer Vision. 2nd ed. Cambridge: Cambridge University Press, 2004. ISBN: 978-0-521-54051-3 (cit. on p. 96).

[He16] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing and SUN, Jian: “Deep Residual Learning for Image Recognition”. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 770–778 (cit. on p. 16).

[He17] HE, Kaiming; GKIOXARI, Georgia; DOLLAR, Piotr and GIRSHICK, Ross: “Mask R-CNN”. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2961–2969 (cit. on p. 17).

[Hel13] HELTEN, Thomas; BAAK, Andreas; BHARAJ, Gaurav; MÜLLER, Meinard; SEIDEL, Hans-Peter and THEOBALT, Christian: “Personalization and Evaluation of a Real-Time Depth-Based Full Body Tracker”. In: *International Conference on 3D Vision (3DV).* June 2013, pp. 279–286 (cit. on p. 19).

[Hes15] Hesse, Nikolas; Stachowiak, Gregor; Breuer, Timo and Arens, Michael: "Estimating Body Pose of Infants in Depth Images Using Random Ferns". In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015, pp. 35–43 (cit. on pp. 18, 19).

[Hir20] Hirsch, Maria; Diederichs, Frederik; Widlroither, Harald; Graf, Ralf and Bischoff, Sven: "Sleep and Take-over in Automated Driving". In: *International Journal of Transportation Science and Technology* 9.1 (Mar. 1, 2020), pp. 42–51. issn: 2046-0430 (cit. on p. 3).

[Hoa16] Hoang Ngan Le, T.; Zheng, Yutong; Zhu, Chenchen; Luu, Khoa and Savvides, Marios: "Multiple Scale Faster-RCNN Approach to Driver's Cell-Phone Usage and Hands on Steering Wheel Detection". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016, pp. 46–53 (cit. on p. 38).

[Hoa17] Hoang Ngan Le, T.; Gia Quach, Kha; Zhu, Chenchen; Nhan Duong, Chi; Luu, Khoa and Savvides, Marios: "Robust Hand Detection and Classification in Vehicles and in the Wild". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017, pp. 39–46 (cit. on p. 26).

[Hoc97] Hochreiter, Sepp and Schmidhuber, Jürgen: "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. issn: 0899-7667 (cit. on pp. 32, 113).

[Hou16] Hou, Yonghong; Li, Zhaoyang; Wang, Pichao and Li, Wanqing: "Skeleton Optical Spectra-Based Action Recognition Using Convolutional Neural Networks". In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.3 (2016), pp. 807–811 (cit. on p. 32).

[Hss17] Hssayeni, Murtadha; Saxena, Sagar; Ptucha, Raymond and Savakis, Andreas: "Distracted Driver Detection: Deep Learning vs Handcrafted Features". In: *Electronic Imaging* 2017 (Jan. 29, 2017), pp. 20–26 (cit. on pp. 39, 40).

[Hu18] Hu, Yaocong; Lu, MingQi and Lu, Xiaobo: "Spatial-Temporal Fusion Convolutional Neural Network for Simulated Driving Behavior Recognition". In: *International Conference on Control, Automation, Robotics and Vision (ICARCV).* Nov. 2018, pp. 1271–1277 (cit. on p. 40).

[Hu19] Hu, Yaocong; Lu, Mingqi and Lu, Xiaobo: "Driving Behaviour Recognition from Still Images by Using Multi-Stream Fusion CNN". In: *Machine Vision and Applications* 30.5 (July 1, 2019), pp. 851–865. issn: 1432-1769 (cit. on p. 39).

[Hu20] Hu, Yaocong; Lu, Mingqi and Lu, Xiaobo: "Feature Refinement for Image-Based Driver Action Recognition via Multi-Scale Attention Convolutional Neural Network". In: *Signal Processing: Image Communication* 81 (Feb. 1, 2020), p. 115697. issn: 0923-5965 (cit. on p. 39).

[Hua17] Huang, Gao; Liu, Zhuang; van der Maaten, Laurens and Weinberger, Kilian Q.: "Densely Connected Convolutional Networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2017, pp. 4700–4708 (cit. on p. 125).

[Hua20] Huang, Zhen; Shen, Xu; Tian, Xinmei; Li, Houqiang; Huang, Jianqiang and Hua, Xian-Sheng: "Spatio-Temporal Inception Graph Convolutional Networks for Skeleton-Based Action Recognition". In: *Proceedings of the 28th ACM International Conference on Multimedia.* New York, NY, USA: Association for Computing Machinery, Oct. 12, 2020, pp. 2122–2130. isbn: 978-1-4503-7988-5 (cit. on pp. 33, 34).

[Ins16] Insafutdinov, Eldar; Pishchulin, Leonid; Andres, Bjoern; Andriluka, Mykhaylo and Schiele, Bernt: "DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model". In: *Computer Vision – ECCV 2016.* Ed. by Leibe, Bastian; Matas, Jiri; Sebe, Nicu and Welling, Max. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 34–50. isbn: 978-3-319-46466-4 (cit. on p. 17).

[Iof15] Ioffe, Sergey and Szegedy, Christian: "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning.* PMLR, June 1, 2015, pp. 448–456 (cit. on p. 30).

[Ion14] Ionescu, C.; Papava, D.; Olaru, V. and Sminchisescu, C.: "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (July 2014), pp. 1325–1339. issn: 1939-3539 (cit. on pp. 15, 25).

[Jai15a] Jain, Arjun; Tompson, Jonathan; LeCun, Yann and Bregler, Christoph: "MoDeep: A Deep Learning Framework Using Motion Features for Human Pose Estimation". In: *Computer Vision – ACCV 2014.* Ed. by Cremers, Daniel; Reid, Ian; Saito, Hideo and Yang, Ming-Hsuan. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 302–315. isbn: 978-3-319-16808-1 (cit. on p. 16).

[Jai15b] Jain, Ashesh; Koppula, Hema S.; Raghavan, Bharad; Soh, Shane and Saxena, Ashutosh: "Car That Knows Before You Do: Anticipating Maneuvers via Learning Temporal Driving Models". In: Proceedings of the IEEE International Conference on Computer Vision. 2015, pp. 3182–3190 (cit. on pp. 36, 37).

[Jam20] Jamsheed V., Abdul; Janet, B. and Reddy, U. Srinivasulu: "Real Time Detection of Driver Distraction Using CNN". In: *International Conference on Smart Systems and Inventive Technology (ICSSIT).* Aug. 2020, pp. 185–191 (cit. on p. 39).

[Jan20] Jang, Jinhyeok; Kim, Dohyung; Park, Cheonshu; Jang, Minsu; Lee, Jaeyeon and Kim, Jaehong: "ETRI-Activity3D: A Large-Scale RGB-D Dataset for Robots to Recognize Daily Activities of the Elderly". In: *International Conference on Intelligent Robots and Systems (IROS).* Oct. 2020, pp. 10990–10997 (cit. on p. 31).

[JEG18] JEGHAM, Imen; BEN KHALIFA, Anouar; ALOUANI, Ihsen and MAHJOUB, Mohamed Ali: "Safe Driving : Driver Action Recognition Using SURF Keypoints". In: *International Conference on Microelectronics (ICM).* Dec. 2018, pp. 60–63 (cit. on p. 39).

[Jeg19] Jegham, Imen; Ben Khalifa, Anouar; Alouani, Ihsen and Mahjoub, Mohamed Ali: "MDAD: A Multimodal and Multiview in-Vehicle Driver Action Dataset". In: *Computer Analysis of Images and Patterns.* Ed. by Vento, Mario and Percannella, Gennaro. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 518–529. isbn: 978-3-030-29888-3 (cit. on pp. 36, 37, 40).

[Jeg20] Jegham, Imen; Ben Khalifa, Anouar; Alouani, Ihsen and Mahjoub, Mohamed Ali: "A Novel Public Dataset for Multimodal Multiview and Multispectral Driver Distraction Analysis: 3MDAD". In: *Signal Processing: Image Communication* 88 (Oct. 1, 2020), p. 115960. issn: 0923-5965 (cit. on pp. 36, 40).

[Jeg21] Jegham, Imen; Khalifa, Anouar Ben; Alouani, Ihsen and Mahjoub, Mohamed Ali: "Soft Spatial Attention-Based Multimodal Driver Action Recognition Using Deep Learning". In: *IEEE Sensors Journal* 21.2 (Jan. 2021), pp. 1918–1925. issn: 1558-1748 (cit. on p. 40).

[Jia21] Jiao, Shuang-Jian; Liu, Lin-Yao and Liu, Qian: "A Hybrid Deep Learning Model for Recognizing Actions of Distracted Drivers". In: *Sensors* 21.21 (21 Jan. 2021), p. 7424. issn: 1424-8220 (cit. on p. 42).

[Jiu14] Jiu, Mingyuan; Wolf, Christian; Taylor, Graham and Baskurt, Atilla: “Human Body Part Estimation from Depth Images via Spatially-Constrained Deep Learning”. In: *Pattern Recognition Letters.* Depth Image Analysis 50 (Dec. 1, 2014), pp. 122–129. issn: 0167-8655 (cit. on p. 20).

[Joh11] Johnson, Sam and Everingham, Mark: “Learning Effective Human Pose Estimation from Inaccurate Annotation”. In: *International Conference on Computer Vision (CVPR).* June 2011, pp. 1465–1472 (cit. on p. 14).

[Joo16] Joo, Hanbyul; Simon, Tomas; Li, Xulong; Liu, Hao et al.: “Panoptic Studio: A Massively Multiview System for Social Interaction Capture”. Dec. 9, 2016. arXiv: 1612.03153 [cs] (cit. on pp. 15, 21, 25).

[Jun16] Jung, Ho Yub; Suh, Yumin; Moon, Gyeongsik and Lee, Kyoung Mu: “A Sequential Approach to 3D Human Pose Estimation: Separation of Localization and Identification of Body Joints”. In: *Computer Vision – ECCV 2016.* Ed. by Leibe, Bastian; Matas, Jiri; Sebe, Nicu and Welling, Max. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 747–761. isbn: 978-3-319-46454-1 (cit. on p. 19).

[Kad20] Kadkhodamohammadi, Abdolrahim and Padoy, Nicolas: “A Generalizable Approach for Multi-View 3D Human Pose Regression”. In: *Machine Vision and Applications* 32.1 (Oct. 8, 2020), p. 6. issn: 1432-1769 (cit. on p. 21).

[Kag16] Kaggle: State Farm Challenge. 2016. url: https://kaggle.com/c/state-farm-distracted-driver-detection (visited on 03/01/2020) (cit. on pp. 35–37).

[Kan18] Kanazawa, Angjoo; Black, Michael J.; Jacobs, David W. and Malik, Jitendra: “End-to-End Recovery of Human Shape and Pose”. In: *Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, 2018 (cit. on p. 22).

[Kap20] Kapoor, Khyati; Pamula, Rajendra and Murthy, Sristi Vns: “Real-Time Driver Distraction Detection System Using Convolutional Neural Networks”. In: *Proceedings of ICETIT 2019.* Ed. by Singh, Pradeep Kumar; Panigrahi, Bijaya Ketan; Suryadevara, Nagender Kumar; Sharma, Sudhir Kumar and Singh, Amit Prakash. Lecture Notes in Electrical Engineering. Cham: Springer International Publishing, 2020, pp. 280–291. isbn: 978-3-030-30577-2 (cit. on p. 39).

[Kat21] Katrolia, Jigyasa Singh; Mirbach, Bruno; El-Sherif, Ahmed; Feld, Hartmut; Rambach, Jason and Stricker, Didier: “TICaM: A Time-of-flight In-car Cabin Monitoring Dataset”. Mar. 23, 2021. arXiv: 2103.11719 [cs] (cit. on pp. 36, 37).

[Ke17] Ke, Qiuhong; Bennamoun, Mohammed; An, Senjian; Sohel, Ferdous and Boussaid, Farid: "A New Representation of Skeleton Sequences for 3D Action Recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 3288–3297 (cit. on p. 32).

[Kin14] Kingma, Diederik P. and Ba, Jimmy: "Adam: A Method for Stochastic Optimization". Dec. 22, 2014. arXiv: 1412.6980 `[cs]` (cit. on p. 116).

[Kip17] Kipf, Thomas N. and Welling, Max: "Semi-Supervised Classification with Graph Convolutional Networks". Feb. 22, 2017. arXiv: 1609.02907 `[cs, stat]` (cit. on pp. 33, 120).

[Koa21] Koay, Hong Vin; Chuah, Joon Huang; Chow, Chee-Onn; Chang, Yang-Lang and Rudrusamy, Bhuvendhraa: "Optimally-Weighted Image-Pose Approach (OWIPA) for Distracted Driver Detection and Classification". In: *Sensors* 21.14 (14 Jan. 2021), p. 4837 (cit. on p. 42).

[Kol10] Kolb, A.; Barth, E.; Koch, R. and Larsen, R.: "Time-of-Flight Cameras in Computer Graphics". In: *Computer Graphics Forum* 29.1 (2010), pp. 141–159. issn: 1467-8659 (cit. on p. 48).

[Kon15] Kondyli, Alexandra; Sisiopiku, Virginia P.; Zhao, Liangke and Barmpoutis, Angelos: "Computer Assisted Analysis of Drivers' Body Activity Using a Range Camera". In: *IEEE Intelligent Transportation Systems Magazine* 7.3 (2015), pp. 18–28. issn: 1941-1197 (cit. on p. 27).

[Kop21] Kopuklu, Okan; Zheng, Jiapeng; Xu, Hang and Rigoll, Gerhard: "Driver Anomaly Detection: A Dataset and Contrastive Learning Approach". In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021, pp. 91–100 (cit. on p. 41).

[Kos19] Kose, Neslihan; Kopuklu, Okan; Unnervik, Alexander and Rigoll, Gerhard: "Real-Time Driver State Monitoring Using a CNN Based Spatio-Temporal Approach". In: *International Conference on Intelligent Transportation Systems (ITSC)*. Oct. 2019, pp. 3236–3242 (cit. on p. 40).

[Kre19] Kreiss, Sven; Bertoni, Lorenzo and Alahi, Alexandre: "PifPaf: Composite Fields for Human Pose Estimation". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 11977–11986 (cit. on pp. 16, 18).

[Kri12] Krizhevsky, Alex; Sutskever, Ilya and Hinton, Geoffrey E: "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012 (cit. on p. 16).

[Kue11] Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T. and Serre, T.: “HMDB: A Large Video Database for Human Motion Recognition”. In: *International Conference on Computer Vision (ICCV).* Nov. 2011, pp. 2556–2563 (cit. on pp. 28, 37).

[Lal14] Lallemand, Joe; Szczot, Magdalena and Ilic, Slobodan: “Human Pose Estimation in Stereo Images”. In: *Articulated Motion and Deformable Objects.* Ed. by Perales, Francisco José and Santos-Victor, José. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 10–19. isbn: 978-3-319-08849-5 (cit. on p. 19).

[Lee19] Leekha, Maitree; Goswami, Mononito; Shah, Rajiv Ratn; Yin, Yifang and Zimmermann, Roger: “Are You Paying Attention? Detecting Distracted Driving in Real-Time”. In: *International Conference on Multimedia Big Data (BigMM).* Sept. 2019, pp. 171–180 (cit. on p. 39).

[Li15] Li, Sijin and Chan, Antoni B.: “3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network”. In: *Computer Vision – ACCV 2014.* Ed. by Cremers, Daniel; Reid, Ian; Saito, Hideo and Yang, Ming-Hsuan. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 332–347. isbn: 978-3-319-16808-1 (cit. on p. 22).

[Li17] Li, Chuankun; Hou, Yonghong; Wang, Pichao and Li, Wanqing: “Joint Distance Maps Based Action Recognition With Convolutional Neural Networks”. In: *IEEE Signal Processing Letters* 24.5 (May 2017), pp. 624–628. issn: 1558-2361 (cit. on p. 32).

[Li18] Li, Peng; Lu, Meiqi and Zhang, Xuetao: “Driver Pose Estimation by Hybrid Convolutional Network Architecture”. In: *Chinese Automation Congress (CAC).* Nov. 2018, pp. 3767–3772 (cit. on p. 27).

[Li19a] Li, Maosen; Chen, Siheng; Chen, Xu; Zhang, Ya; Wang, Yanfeng and Tian, Qi: “Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, pp. 3595–3603 (cit. on pp. 33, 34).

[Li19b] Li, Peng; Lu, Meiqi; Zhang, Zhiwei; Shan, Donghui and Yang, Yang: “A Novel Spatial-Temporal Graph for Skeleton-based Driver Action Recognition”. In: *International Conference on Intelligent Transportation Systems (ITSC).* Oct. 2019, pp. 3243–3248 (cit. on p. 42).

[Li20] Li, Liam; Jamieson, Kevin; Rostamizadeh, Afshin; Gonina, Ekaterina; Hardt, Moritz; Recht, Benjamin and Talwalkar, Ameet: A System for Massively Parallel Hyperparameter Tuning. Mar. 15, 2020. arXiv: 1810.05934 [cs, stat] (cit. on p. 116).

[Lia18] Liaw, Richard; Liang, Eric; Nishihara, Robert; Moritz, Philipp; Gonzalez, Joseph E. and Stoica, Ion: Tune: A Research Platform for Distributed Model Selection and Training. July 13, 2018. arXiv: 1807.05118 [cs, stat] (cit. on p. 116).

[Lin14] Lin, Tsung-Yi; Maire, Michael; Belongie, Serge; Hays, James; Perona, Pietro; Ramanan, Deva; Dollár, Piotr and Zitnick, C. Lawrence: "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by Fleet, David; Pajdla, Tomas; Schiele, Bernt and Tuytelaars, Tinne. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 740–755. isbn: 978-3-319-10602-1 (cit. on pp. 14, 16, 25, 73).

[Lin18] Lin, Rui; Ma, Liang and Zhang, Wei: "An Interview Study Exploring Tesla Drivers' Behavioural Adaptation". In: *Applied Ergonomics* 72 (Oct. 1, 2018), pp. 37–47. issn: 0003-6870 (cit. on p. 2).

[Lin21] Lin, Zeyang; Liu, Yinchuan and Zhang, Xuetao: "Driver-Skeleton: A Dataset for Driver Action Recognition". In: *International Conference on Intelligent Transportation Systems (ITSC)*. Sept. 2021, pp. 1509–1514 (cit. on pp. 36, 37, 43).

[Liu13] Liu, Gang; Yan, Xinping and Sun, Yufen: "Driver Pose Estimation Using a Mixture-model Method". In: *Proceedings of the Second International Conference on Innovative Computing and Cloud Computing*. ICCC '13. New York, NY, USA: Association for Computing Machinery, Dec. 1, 2013, pp. 200–204. isbn: 978-1-4503-2119-8 (cit. on p. 27).

[Liu16] Liu, Jun; Shahroudy, Amir; Xu, Dong and Wang, Gang: "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition". In: *Computer Vision – ECCV 2016*. Ed. by Leibe, Bastian; Matas, Jiri; Sebe, Nicu and Welling, Max. Lecture Notes in Computer Science. Springer International Publishing, 2016, pp. 816–833. isbn: 978-3-319-46487-9 (cit. on pp. 32, 33, 115).

[Liu17a] Liu, Jun; Wang, Gang; Hu, Ping; Duan, Ling-Yu and Kot, Alex C.: "Global Context-Aware Attention LSTM Networks for 3D Action Recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 1647–1656 (cit. on p. 31).

[Liu17b] Liu, Mengyuan; Liu, Hong and Chen, Chen: "Enhanced Skeleton Visualization for View Invariant Human Action Recognition". In: *Pattern Recognition* 68 (Aug. 1, 2017), pp. 346–362. issn: 0031-3203 (cit. on p. 32).

[Liu19a] Liu, Jun; Shahroudy, Amir; Perez, Mauricio; Wang, Gang; Duan, Ling-Yu and Kot, Alex C.: "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–1. issn: 0162-8828, 2160-9292, 1939-3539. arXiv: 1905.04757 (cit. on pp. 19, 29, 30, 37).

[Liu19b] Liu, Yazhou; Lasang, Pongsak; Pranata, Sugiri; Shen, Shengmei and Zhang, Wenchao: "Driver Pose Estimation Using Recurrent Lightweight Network and Virtual Data Augmented Transfer Learning". In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (Oct. 2019), pp. 3818–3831. issn: 1558-0016 (cit. on p. 27).

[Liu20] Liu, Ziyu; Zhang, Hongwen; Chen, Zhenghao; Wang, Zhiyong and Ouyang, Wanli: "Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 143–152 (cit. on p. 33).

[Liu21] Liu, Dichao; Yamasaki, Toshihiko; Wang, Yu; Mase, Kenji and Kato, Jien: "TML: A Triple-Wise Multi-Task Learning Framework for Distracted Driver Recognition". In: *IEEE Access* 9 (2021), pp. 125955–125969. issn: 2169-3536 (cit. on p. 41).

[Lot19] Lotz, Alexander and Weissenberger, Sarah: "Predicting Take-Over Times of Truck Drivers in Conditional Autonomous Driving". In: *Advances in Human Aspects of Transportation.* Ed. by Stanton, Neville. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2019, pp. 329–338. isbn: 978-3-319-93885-1 (cit. on p. 6).

[Lu19] Lu, Mingqi; Hu, Yaocong and Lu, Xiaobo: "Dilated Light-Head R-CNN Using Tri-Center Loss for Driving Behavior Recognition". In: *Image and Vision Computing* 90 (Oct. 1, 2019), p. 103800. issn: 0262-8856 (cit. on p. 39).

[Lud18] *Ludwig, Julian; *Martin, Manuel; Horne, Matthias; Flad, Michael; Voit, Michael; Stiefelhagen, Rainer and Hohmann, Sören: "Driver Observation and Shared Vehicle Control: Supporting the Driver on the Way Back into the Control Loop". In: *at - Automatisierungstechnik* 66.2 (2018), pp. 146–159. issn: 0178-2312 (cit. on pp. 4, 5).

[Lud21] Ludwig, Julian: Automatisierte kooperative Transition einer Regelungsaufgabe zwischen Mensch und Maschine am Beispiel des hochautomatisierten

Fahrens. Karlsruher Beiträge zur Regelungs- und Steuerungstechnik / Karlsruher Institut für Technologie, Institut für Regelungs- und Steuerungssysteme 9. KIT Scientific Publishing, 2021. ISBN: 978-3-7315-1069-7 (cit. on p. 7).

[Maf20] MAFENI MASE, Jimiama; CHAPMAN, Peter; FIGUEREDO, Grazziela P. and TORRES TORRES, Mercedes: “A Hybrid Deep Learning Approach for Driver Distraction Detection”. In: *International Conference on Information and Communication Technology Convergence (ICTC).* Oct. 2020, pp. 1–6 (cit. on p. 39).

[Mag19] MAGHOUMI, Mehran and LAVIOLA, Joseph J.: “DeepGRU: Deep Gesture Recognition Utility”. In: *Advances in Visual Computing.* Ed. by BEBIS, George; BOYLE, Richard; PARVIN, Bahram; KORACIN, Darko et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 16–31. ISBN: 978-3-030-33720-9 (cit. on p. 31).

[Maj18] MAJDI, Mohammed S.; RAM, Sundaresh; GILL, Jonathan T. and RODRÍGUEZ, Jeffrey J.: “Drive-Net: Convolutional Network for Driver Distraction Detection”. In: *Southwest Symposium on Image Analysis and Interpretation (SSIAI).* Apr. 2018, pp. 1–4 (cit. on p. 39).

[Man20] MANSTETTEN, Dietrich; BERUSCHA, Frank; BIEG, Hans-Joachim; KOBIELA, Fanny; KORTHAUER, Andreas; KRAUTTER, Wolfgang and MARBERGER, Claus: “The Evolution of Driver Monitoring Systems: A Shortened Story on Past, Current and Future Approaches How Cars Acquire Knowledge About the Driver’s State”. In: *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services.* MobileHCI ’20. New York, NY, USA: Association for Computing Machinery, Oct. 5, 2020, pp. 1–6. ISBN: 978-1-4503-8052-2 (cit. on p. 7).

[Mar16] MARTIN, Manuel; LI, Kangxiong; VOIT, Michael; MELCHER, Vivien; WIDLROITHER, Harald; DIEDERICHS, Frederik and STIEFELHAGEN, Rainer: “Klassifikation von Fahrerzuständen Und Nebentätigkeiten Über Körperposen Bei Automatisierter Fahrt”. In: *32. VDI/VW Gemeinschaftstagung ”Fahrerassistenzsysteme Und Automatisiertes Fahren”.* 2016 (cit. on pp. 11, 27, 44).

[Mar17a] MARTIN, Manuel; STUEHMER, Stephan; VOIT, Michael and STIEFELHAGEN, Rainer: “Real Time Driver Body Pose Estimation for Novel Assistance Systems”. In: *International Conference on Intelligent Transportation Systems (ITSC).* 2017, pp. 1–7 (cit. on pp. 10, 11, 23–25, 27, 38, 43, 44, 79, 108, 120).

[Mar17b] Martinez, Julieta; Hossain, Rayat; Romero, Javier and Little, James J.: "A Simple yet Effective Baseline for 3D Human Pose Estimation". In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2640–2649 (cit. on pp. 22, 27, 94).

[Mar18a] Marín-Jiménez, Manuel J.; Romero-Ramirez, Francisco J.; Muñoz-Salinas, Rafael and Medina-Carnicer, Rafael: "3D Human Pose Estimation from Depth Maps Using a Deep Combination of Poses". In: *Journal of Visual Communication and Image Representation* 55 (Aug. 1, 2018), pp. 627–639. issn: 1047-3203 (cit. on p. 20).

[Mar18b] Martin, Manuel; Popp, Johannes; Anneken, Mathias; Voit, Michael and Stiefelhagen, Rainer: "Body Pose and Context Information for Driver Secondary Task Detection". In: *Intelligent Vehicles Symposium (IV).* 2018, pp. 2015–2021 (cit. on pp. 10, 12, 36, 37, 43, 44, 112, 118, 134).

[Mar18c] Martínez-González, Angel; Villamizar, Michael; Canévet, Olivier and Odobez, Jean-Marc: "Real-Time Convolutional Networks for Depth-based Human Pose Estimation". In: *International Conference on Intelligent Robots and Systems (IROS).* Oct. 2018, pp. 41–47 (cit. on pp. 16, 20).

[Mar19] *Martin, Manuel; *Roitberg, Alina; Haurilet, Monica; Horne, Matthias; Reiss, Simon; Voit, Michael and Stiefelhagen, Rainer: "Drive&Act: A Multi-Modal Dataset for Fine-Grained Driver Behavior Recognition in Autonomous Vehicles". In: *International Conference on Computer Vision (ICCV).* 2019, pp. 2801–2810 (cit. on pp. 10, 34, 36, 37, 40, 43, 44, 61, 118).

[Mar20a] Martin, Manuel; Voit, Michael and Stiefelhagen, Rainer: "Dynamic Interaction Graphs for Driver Activity Recognition". In: *International Conference on Intelligent Transportation Systems (ITSC).* 2020 (cit. on pp. 10, 12, 36, 43, 44, 61, 71, 134).

[Mar20b] Martínez-González, Angel; Villamizar, Michael; Canévet, Olivier and Odobez, Jean-Marc: "Residual Pose: A Decoupled Approach for Depth-based 3D Human Pose Estimation". Nov. 10, 2020. arXiv: 2011.05010 [cs] (cit. on p. 20).

[Mar21] Martin, Manuel; Voit, Michael and Stiefelhagen, Rainer: "An Evaluation of Different Methods for 3D-Driver-Body-Pose Estimation". In: *International Conference on Intelligent Transportation Systems (ITSC).* 2021 (cit. on pp. 10, 11, 23, 25, 26, 28, 44, 61, 70, 92).

[Mas18] Masood, Sarfaraz; Rai, Abhinav; Aggarwal, Aakash; Doja, M. N. and Ahmad, Musheer: “Detecting Distraction of Drivers Using Convolutional Neural Network”. In: *Pattern Recognition Letters* (Jan. 12, 2018). issn: 0167-8655 (cit. on p. 39).

[Meh17a] Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W. and Theobalt, C.: “Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision”. In: *International Conference on 3D Vision (3DV).* Oct. 2017, pp. 506–516 (cit. on pp. 15, 74).

[Meh17b] Mehta, Dushyant; Sridhar, Srinath; Sotnychenko, Oleksandr; Rhodin, Helge; Shafiei, Mohammad; Seidel, Hans-Peter; Xu, Weipeng; Casas, Dan and Theobalt, Christian: “VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera”. In: *ACM Trans. Graph.* 36.4 (July 2017), 44:1–44:14. issn: 0730-0301 (cit. on p. 22).

[Mer12] Merat, Natasha; Jamson, A. Hamish; Lai, Frank C. H. and Carsten, Oliver: “Highly Automated Driving, Secondary Task Performance, and Driver State”. In: *Human Factors* 54.5 (Oct. 1, 2012), pp. 762–771. issn: 0018-7208 (cit. on pp. 4, 5).

[Mer14] Merat, Natasha; Jamson, A. Hamish; Lai, Frank C. H.; Daly, Michael and Carsten, Oliver M. J.: “Transition to Manual: Driver Behaviour When Resuming Control from a Highly Automated Vehicle”. In: *Transportation Research Part F: Traffic Psychology and Behaviour.* Vehicle Automation and Driver Behaviour 27 (Nov. 1, 2014), pp. 274–282. issn: 1369-8478 (cit. on p. 5).

[Mol15] Molchanov, Pavlo; Gupta, Shalini; Kim, Kihwan and Kautz, Jan: “Hand Gesture Recognition With 3D Convolutional Neural Networks”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015, pp. 1–7 (cit. on p. 26).

[Moo18] Moon, Gyeongsik; Yong Chang, Ju and Mu Lee, Kyoung: “V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation From a Single Depth Map”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 5079–5088 (cit. on p. 20).

[Moo19] Moon, Gyeongsik; Chang, Ju Yong and Lee, Kyoung Mu: “PoseFix: Model-agnostic General Human Pose Refinement Network”. Mar. 10, 2019. arXiv: 1812.03595 [cs] (cit. on p. 94).

[Mos19] Moslemi, Negar; Azmi, Reza and Soryani, Mohsen: “Driver Distraction Recognition Using 3D Convolutional Neural Networks”. In: *International Conference on Pattern Recognition and Image Analysis (IPRIA).* Mar. 2019, pp. 145–151 (cit. on pp. 40, 41).

[Mur17] Murthy, Pramod Narasimha; Kovalenko, Onorina; Elhayek, Ahmed; Couto Gava, Christiano and Stricker, Didier: “3D Human Pose Tracking inside Car Using Single RGB Spherical Camera”. In: *ACM Chapters Computer Science in Cars Symposium CSCS 2017 |. ACM Chapters Computer Science in Cars Symposium (CSCS-17), July 6, Munich, Germany.* ACM, July 2017 (cit. on p. 23).

[Nat18] National Highway Safety Board (NTSB): Rear-End Collision Between a Car Operating with Advanced Driver Assistance Systems and a Stationary Fire Truck, Culver City, California, January 22, 2018 (cit. on p. 2).

[Nav21] Naveed, Humza; Jafri, Fareed; Javed, Kashif and Babri, Haroon Atique: “Driver Activity Recognition by Learning Spatiotemporal Features of Pose and Human Object Interaction”. In: *Journal of Visual Communication and Image Representation* 77 (May 1, 2021), p. 103135. issn: 1047-3203 (cit. on pp. 36, 37).

[Nel21] Nel, Francois and Ngxande, Mkhuseli: “Driver Activity Recognition Through Deep Learning”. In: *Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA).* Jan. 2021, pp. 1–6 (cit. on p. 40).

[New16] Newell, Alejandro; Yang, Kaiyu and Deng, Jia: “Stacked Hourglass Networks for Human Pose Estimation”. In: *Computer Vision – ECCV 2016.* Ed. by Leibe, Bastian; Matas, Jiri; Sebe, Nicu and Welling, Max. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 483–499. isbn: 978-3-319-46484-8 (cit. on p. 16).

[Ngu21] Nguyen, Duy-Linh; Dwisnanto Putro, Muhamad; Vo, Xuan-Thuy and Jo, Kang-Hyun: “Light-Weight Convolutional Neural Network for Distracted Driver Classification”. In: *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society.* Oct. 2021, pp. 1–6 (cit. on p. 39).

[Nor21] Nordhoff, Sina; Stapel, Jork; He, Xiaolin; Gentner, Alexandre and Happee, Riender: “Perceived Safety and Trust in SAE Level 2 Partially Automated Cars: Results from an Online Questionnaire”. In: *PLOS ONE* 16.12 (Dec. 21, 2021), e0260953. issn: 1932-6203 (cit. on p. 2).

[Ofl13] Ofli, Ferda; Chaudhry, Rizwan; Kurillo, Gregorij; Vidal, René and Bajcsy, Ruzena: "Berkeley MHAD: A Comprehensive Multimodal Human Action Database". In: *Workshop on Applications of Computer Vision (WACV).* Jan. 2013, pp. 53–60 (cit. on p. 15).

[Ohn13] Ohn-Bar, Eshed and Trivedi, Mohan: "In-Vehicle Hand Activity Recognition Using Integration of Regions". In: *Intelligent Vehicles Symposium (IV).* June 2013, pp. 1034–1039 (cit. on pp. 38, 111).

[Oku18] Okuno, Kaoruko; Yamashita, Takayoshi; Fukui, Hiroshi; Noridomi, Shuzo; Arata, Koji; Yamauchi, Yuji and Fujiyoshi, Hironobu: "Body Posture and Face Orientation Estimation by Convolutional Network with Heterogeneous Learning". In: *International Workshop on Advanced Image Technology (IWAIT).* Jan. 2018, pp. 1–4 (cit. on p. 27).

[Ort20] Ortega, Juan Diego; Kose, Neslihan; Cañas, Paola; Chao, Min-An; Unnervik, Alexander; Nieto, Marcos; Otaegui, Oihana and Salgado, Luis: "DMD: A Large-Scale Multi-modal Driver Monitoring Dataset for Attention and Alertness Analysis". In: *Computer Vision – ECCV 2020 Workshops.* Ed. by Bartoli, Adrien and Fusiello, Andrea. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 387–405. isbn: 978-3-030-66823-5 (cit. on pp. 36, 37, 40).

[Pan21] Pan, Chaopeng; Cao, Haotian; Zhang, Weiwei; Song, Xiaolin and Li, Mingjun: "Driver Activity Recognition Using Spatial-Temporal Graph Convolutional LSTM Networks with Attention Mechanism". In: *IET Intelligent Transport Systems* 15.2 (Feb. 2021). issn: 1751-956X (cit. on p. 42).

[Pap17] Papandreou, George; Zhu, Tyler; Kanazawa, Nori; Toshev, Alexander; Tompson, Jonathan; Bregler, Chris and Murphy, Kevin: "Towards Accurate Multi-Person Pose Estimation in the Wild". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 4903–4911 (cit. on pp. 16, 17).

[Pap18] Papandreou, George; Zhu, Tyler; Chen, Liang-Chieh; Gidaris, Spyros; Tompson, Jonathan and Murphy, Kevin: "PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model". In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 269–286 (cit. on p. 18).

[Per17] Perrett, Toby; Mirmehdi, Majid and Dias, Eduardo: "Visual Monitoring of Driver and Passenger Control Panel Interactions". In: *IEEE Transactions on Intelligent Transportation Systems* 18.2 (Feb. 2017), pp. 321–331. issn: 1558-0016 (cit. on p. 38).

[Pet13] Petermann-Stock, Ina; Hackenberg, Linn; Muhr, Tobias and Mergl, Christian: “Wie Lange Braucht Der Fahrer? Eine Analyse Zu Übernahmezeiten Aus Verschiedenen Nebentätigkeiten Während Einer Hochautomatisierten Staufahrt”. In: *6. Tagung Fahrerassistenzsysteme. Der Weg zum automatischen Fahren* (2013) (cit. on pp. 4, 5).

[Pis16] Pishchulin, Leonid; Insafutdinov, Eldar; Tang, Siyu; Andres, Bjoern; Andriluka, Mykhaylo; Gehler, Peter V. and Schiele, Bernt: “DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 4929–4937 (cit. on p. 17).

[Pur17] Purington, Amanda; Taft, Jessie G.; Sannon, Shruti; Bazarova, Natalya N. and Taylor, Samuel Hardman: “”Alexa Is My New BFF”: Social Roles, User Satisfaction, and Personification of the Amazon Echo”. In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’17. New York, NY, USA: Association for Computing Machinery, May 6, 2017, pp. 2853–2859. isbn: 978-1-4503-4656-6 (cit. on p. 9).

[Qin21] Qin, Binbin; Qian, Jiangbo; Xin, Yu; Liu, Baisong and Dong, Yihong: “Distracted Driver Detection Based on a CNN With Decreasing Filter Size”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–12. issn: 1558-0016 (cit. on p. 39).

[Qiu17] Qiu, Zhaofan; Yao, Ting and Mei, Tao: “Learning Spatio-Temporal Representation With Pseudo-3D Residual Networks”. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 5533–5541 (cit. on pp. 30, 134).

[Qui09] Quigley, Morgan; Conley, Ken; Gerkey, Brian; Faust, Josh; Foote, Tully; Leibs, Jeremy; Wheeler, Rob and Ng, Andrew: “ROS: An Open-Source Robot Operating System”. In: *ICRA Workshop on Open Source Software*. Vol. 3. Jan. 2009 (cit. on p. 63).

[Rad14] Radlmayr, Jonas; Gold, Christian; Lorenz, Lutz; Farid, Mehdi and Bengler, Klaus: “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*. Vol. 58. 2014, pp. 2063–2067 (cit. on pp. 4, 5).

[Raf15] Rafi, Umer; Gall, Juergen and Leibe, Bastian: "A Semantic Occlusion Model for Human Pose Estimation From a Single Depth Image". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015, pp. 67–74 (cit. on pp. 18, 19).

[Raj18] Raja, Muneeba; Ghaderi, Viviane and Sigg, Stephan: "WiBot! In-Vehicle Behaviour and Gesture Recognition Using Wireless Network Edge". In: *International Conference on Distributed Computing Systems (ICDCS).* July 2018, pp. 376–387 (cit. on p. 8).

[Ran18] Rangesh, Akshay and Trivedi, Mohan M.: "HandyNet: A One-Stop Solution to Detect, Segment, Localize & Analyze Driver Hands". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018, pp. 1103–1110 (cit. on p. 26).

[Red13] Reddy, Kishore K. and Shah, Mubarak: "Recognizing 50 Human Action Categories of Web Videos". In: *Machine Vision and Applications* 24.5 (July 1, 2013), pp. 971–981. issn: 1432-1769 (cit. on p. 28).

[Rei20a] Reiß, Simon; Roitberg, Alina; Haurilet, Monica and Stiefelhagen, Rainer: "Deep Classification-driven Domain Adaptation for Cross-Modal Driver Behavior Recognition". In: *Intelligent Vehicles Symposium (IV).* Oct. 2020, pp. 1042–1047 (cit. on pp. 41, 138, 139).

[Rei20b] Reiss, Simon; Roitberg, Alina; Haurilet, Monica and Stiefelhagen, Rainer: "Activity-Aware Attributes for Zero-Shot Driver Behavior Recognition". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020, pp. 902–903 (cit. on pp. 36, 41, 44).

[Ren21] Ren, Hongze; Guo, Yage; Bai, Zhonghao and Cheng, Xiangyu: "A Multi-Semantic Driver Behavior Recognition Model of Autonomous Vehicles Using Confidence Fusion Mechanism". In: *Actuators* 10.9 (9 Sept. 2021), p. 218. issn: 2076-0825 (cit. on p. 41).

[Rib21] Ribas, Luis Gustavo Tomal; Cocron, Marta Pereira; Da Silva, Joed Lopes; Zimmer, Alessandro and Brandmeier, Thomas: "In-Cabin Vehicle Synthetic Data to Test Deep Learning Based Human Pose Estimation Models". In: *Intelligent Vehicles Symposium (IV).* July 2021, pp. 610–615 (cit. on p. 52).

[Roi20a] Roitberg, Alina; Haurilet, Monica; Reiß, Simon and Stiefelhagen, Rainer: "CNN-based Driver Activity Understanding: Shedding Light on Deep Spatiotemporal Representations". In: *International Conference on Intelligent Transportation Systems (ITSC).* Sept. 2020, pp. 1–6 (cit. on p. 41).

[Roi20b] Roitberg, Alina; Ma, Chaoxiang; Haurilet, Monica and Stiefelhagen, Rainer: “Open Set Driver Activity Recognition”. In: *Intelligent Vehicles Symposium (IV).* Oct. 2020, pp. 1048–1053 (cit. on pp. 36, 41, 44).

[Roi21] Roitberg, Alina; Haurilet, Monica; Martinez, Manuel and Stiefelhagen, Rainer: “Uncertainty-Sensitive Activity Recognition: A Reliability Benchmark and the CARING Models”. In: *International Conference on Pattern Recognition (ICPR).* Jan. 2021, pp. 3814–3821 (cit. on p. 41).

[SAE21] SAE International: “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles”. In: *Ground Vehicle Standard J3016_202104* (Apr. 30, 2021) (cit. on p. 2).

[Sap13] Sapp, Ben and Taskar, Ben: “MODEC: Multimodal Decomposable Models for Human Pose Estimation”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013, pp. 3674–3681 (cit. on p. 14).

[Sch15] Schwarz, Anke; Wacker, Esther-Sabrina; Martin, Manuel; Sarfraz, M. Saquib and Stiefelhagen, Rainer: “3D Facial Landmark Detection: How to Deal with Head Rotations?” In: *Pattern Recognition.* Ed. by Gall, Juergen; Gehler, Peter and Leibe, Bastian. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 424–434. isbn: 978-3-319-24947-6 (cit. on p. 26).

[Sch17] Schwarz, Anke; Haurilet, Monica; Martinez, Manuel and Stiefelhagen, Rainer: “DriveAHead - A Large-Scale Driver Head Pose Dataset”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2017, pp. 1–10 (cit. on p. 26).

[Sch18] Schwarz, Anke: Tiefen-basierte Bestimmung der Kopfposition und -orientierung im Fahrzeuginnenraum. Karlsruher Institut für Technologie (KIT), 2018 (cit. on p. 48).

[Sch20] Schmidt, Eike A.; Kuiper, Ouren X.; Wolter, Stefan; Diels, Cyriel and Bos, Jelte E.: “An International Survey on the Incidence and Modulating Factors of Carsickness”. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 71 (May 1, 2020), pp. 76–87. issn: 1369-8478 (cit. on p. 8).

[Sha16a] Shafaei, Alireza and Little, James J.: “Real-Time Human Motion Capture with Multiple Depth Cameras”. In: *Conference on Computer and Robot Vision (CRV).* June 2016, pp. 24–31 (cit. on pp. 16, 20).

[Sha16b] Shahroudy, Amir; Liu, Jun; Ng, Tian-Tsong and Wang, Gang: "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 1010–1019 (cit. on pp. 19, 29–31, 33, 37).

[Shi14] Shia, Victor A.; Gao, Yiqi; Vasudevan, Ramanarayan; Campbell, Katherine Driggs; Lin, Theresa; Borrelli, Francesco and Bajcsy, Ruzena: "Semiautonomous Vehicular Control Using Driver Modeling". In: *IEEE Transactions on Intelligent Transportation Systems* 15.6 (Dec. 2014), pp. 2696–2709. issn: 1558-0016 (cit. on p. 26).

[Shi19a] Shi, Lei; Zhang, Yifan; Cheng, Jian and Lu, Hanqing: "Skeleton-Based Action Recognition With Directed Graph Neural Networks". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* 2019, pp. 7912–7921 (cit. on pp. 31, 34).

[Shi19b] Shi, Lei; Zhang, Yifan; Cheng, Jian and Lu, Hanqing: "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition". July 9, 2019. arXiv: 1805.07694 [cs] (cit. on p. 34).

[Shi20] Shi, Lei; Zhang, Yifan; Cheng, Jian and Lu, Hanqing: "Skeleton-Based Action Recognition With Multi-Stream Adaptive Graph Convolutional Networks". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9532–9545. issn: 1941-0042 (cit. on pp. 31, 34).

[Sho11] Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A. and Blake, A.: "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* June 2011, pp. 1297–1304 (cit. on pp. 18, 19, 77, 79, 81, 87, 93).

[Sho13] Shotton, Jamie; Girshick, Ross; Fitzgibbon, Andrew; Sharp, Toby et al.: "Efficient Human Pose Estimation from Single Depth Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.12 (Dec. 2013), pp. 2821–2840. issn: 1939-3539 (cit. on p. 19).

[Sid16] Siddharth; Rangesh, Akshay; Ohn-Bar, Eshed and Trivedi, Mohan M.: "Driver Hand Localization and Grasp Analysis: A Vision-Based Real-Time Approach". In: *International Conference on Intelligent Transportation Systems (ITSC).* Nov. 2016, pp. 2545–2550 (cit. on p. 38).

[Sig10] Sigal, Leonid; Balan, Alexandru O and Black, Michael J: "Humaneva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion". In: *International journal of computer vision* 87.1-2 (2010), p. 4 (cit. on p. 15).

[Sim14] Simonyan, Karen and Zisserman, Andrew: “Two-Stream Convolutional Networks for Action Recognition in Videos”. In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014 (cit. on p. 30).

[Son17a] Song, Sijie; Lan, Cuiling; Xing, Junliang; Zeng, Wenjun and Liu, Jiaying: “An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (1 Feb. 12, 2017). issn: 2374-3468 (cit. on p. 31).

[Son17b] Sonnenberg, Jan and Fliegner, Jens: InCarIn - IntelligentCarInterieur : Insassen- Und Innenraumkontextanalyse Im Automobil Zur Adaptiven Und Kontextsensitiven Anpassung von Interaktionstechnik Und Assistenzsysteme Für Alle. [Wolfsburg]: [Volkswagen AG], 2017 (cit. on p. 55).

[Soo12] Soomro, Khurram; Zamir, Amir Roshan and Shah, Mubarak: “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild”. Dec. 3, 2012. arXiv: 1212.0402 [cs] (cit. on p. 28).

[Sri21] Srivastav, Vinkle; Issenhuth, Thibaut; Kadkhodamohammadi, Abdolrahim; de Mathelin, Michel; Gangi, Afshin and Padoy, Nicolas: “MVOR: A Multi-view RGB-D Operating Room Dataset for 2D and 3D Human Pose Estimation”. Aug. 20, 2021. arXiv: 1808.08180 [cs] (cit. on p. 15).

[Sun18] Sun, Xiao; Xiao, Bin; Wei, Fangyin; Liang, Shuang and Wei, Yichen: “Integral Human Pose Regression”. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 529–545 (cit. on p. 16).

[Sun19] Sun, Ke; Xiao, Bin; Liu, Dong and Wang, Jingdong: “Deep High-Resolution Representation Learning for Human Pose Estimation”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, pp. 5693–5703 (cit. on p. 16).

[Sun21] Sun, Zehua; Liu, Jun; Ke, Qiuhong; Rahmani, Hossein; Bennamoun, Mohammed and Wang, Gang: “Human Action Recognition from Various Data Modalities: A Review”. Jan. 29, 2021. arXiv: 2012.11866 [cs] (cit. on p. 29).

[Tan21] Tan, Mingkui; Ni, Gengqin; Liu, Xu; Zhang, Shiliang; Wu, Xiangmiao; Wang, Yaowei and Zeng, Runhao: “Bidirectional Posture-Appearance Interaction Network for Driver Behavior Recognition”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–13. issn: 1558-0016 (cit. on pp. 36, 37, 42, 134).

[Tay12] Taylor, Jonathan; Shotton, Jamie; Sharp, Toby and Fitzgibbon, Andrew: "The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation". In: *Conference on Computer Vision and Pattern Recognition (CVPR).* June 2012, pp. 103–110 (cit. on p. 19).

[Thi87] Thibault, William C. and Naylor, Bruce F.: "Set Operations on Polyhedra Using Binary Space Partitioning Trees". In: *Proceedings of the Conference on Computer Graphics and Interactive Techniques (SIGGRAPH).* 1987, pp. 153–162 (cit. on p. 72).

[Tom12] Toma, Madalina-Ioana; Rothkrantz, Leon J.M. and Antonya, Csaba: "Car Driver Skills Assessment Based on Driving Postures Recognition". In: *International Conference on Cognitive Infocommunications (CogInfoCom).* Dec. 2012, pp. 439–446 (cit. on p. 26).

[Tom14] Tompson, Jonathan J; Jain, Arjun; LeCun, Yann and Bregler, Christoph: "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation". In: *Advances in Neural Information Processing Systems.* Vol. 27. Curran Associates, Inc., 2014 (cit. on p. 16).

[Tor19] Torres, Helena R.; Oliveira, Bruno; Fonseca, Jaime; Queirós, Sandro; Borges, João; Rodrigues, Nélson; Coelho, Victor; Pallauf, Johannes; Brito, José and Mendes, José: "Real-Time Human Body Pose Estimation for In-Car Depth Images". In: *Technological Innovation for Industry and Service Systems.* Ed. by Camarinha-Matos, Luis M.; Almeida, Ricardo and Oliveira, José. IFIP Advances in Information and Communication Technology. Cham: Springer International Publishing, 2019, pp. 169–182. isbn: 978-3-030-17771-3 (cit. on p. 27).

[Tos14] Toshev, Alexander and Szegedy, Christian: "DeepPose: Human Pose Estimation via Deep Neural Networks". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014, pp. 1653–1660 (cit. on p. 16).

[Tra10] Tran, Cuong and Trivedi, Mohan M.: "Towards a Vision-Based System Exploring 3D Driver Posture Dynamics for Driver Assistance: Issues and Possibilities". In: *Intelligent Vehicles Symposium (IV).* June 2010, pp. 179–184 (cit. on p. 27).

[Tra15] Tran, Du; Bourdev, Lubomir; Fergus, Rob; Torresani, Lorenzo and Paluri, Manohar: "Learning Spatiotemporal Features With 3D Convolutional Networks". In: Proceedings of the IEEE International Conference on Computer Vision. 2015, pp. 4489–4497 (cit. on pp. 30, 134).

[Tu20] Tu, Hanyue; Wang, Chunyu and Zeng, Wenjun: “VoxelPose: Towards Multi-Camera 3D Human Pose Estimation in Wild Environment”. Aug. 24, 2020. arXiv: 2004.06239 [cs] (cit. on p. 21).

[Val18] Valeriano, Leonel Cuevas; Napoletano, Paolo and Schettini, Raimondo: “Recognition of Driver Distractions Using Deep Learning”. In: *International Conference on Consumer Electronics - Berlin (ICCE-Berlin).* Sept. 2018, pp. 1–6 (cit. on p. 40).

[Var17] Varol, Gul; Romero, Javier; Martin, Xavier; Mahmood, Naureen; Black, Michael J.; Laptev, Ivan and Schmid, Cordelia: “Learning From Synthetic Humans”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 109–117 (cit. on p. 16).

[Vee15] Veeriah, Vivek; Zhuang, Naifan and Qi, Guo-Jun: “Differential Recurrent Neural Networks for Action Recognition”. In: Proceedings of the IEEE International Conference on Computer Vision. 2015, pp. 4041–4049 (cit. on p. 33).

[vMar18] Von Marcard, Timo; Henschel, Roberto; Black, Michael J.; Rosenhahn, Bodo and Pons-Moll, Gerard: “Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera”. In: Proceedings of the European Conference on Computer Vision (ECCV). 2018, pp. 601–617 (cit. on p. 16).

[Vor18] Vora, Sourabh; Rangesh, Akshay and Trivedi, Mohan Manubhai: “Driver Gaze Zone Estimation Using Convolutional Neural Networks: A General Framework and Ablative Analysis”. In: *IEEE Transactions on Intelligent Vehicles* 3.3 (Sept. 2018), pp. 254–265. issn: 2379-8858 (cit. on p. 6).

[Wan15] Wang, Qifei; Kurillo, Gregorij; Ofli, Ferda and Bajcsy, Ruzena: “Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect”. In: *International Conference on Healthcare Informatics.* Oct. 2015, pp. 380–389 (cit. on p. 19).

[Wan16a] Wang, Keze; Zhai, Shengfu; Cheng, Hui; Liang, Xiaodan and Lin, Liang: “Human Pose Estimation from Depth Images via Inference Embedded Multi-task Learning”. In: *Proceedings of the 24th ACM International Conference on Multimedia.* MM ’16. New York, NY, USA: Association for Computing Machinery, Oct. 1, 2016, pp. 1227–1236. isbn: 978-1-4503-3603-1 (cit. on pp. 15, 20).

[Wan16b] WANG, Pichao; LI, Zhaoyang; HOU, Yonghong and LI, Wanqing: "Action Recognition Based on Joint Trajectory Maps Using Convolutional Neural Networks". In: *Proceedings of the 24th ACM International Conference on Multimedia.* MM '16. Amsterdam, The Netherlands: Association for Computing Machinery, Oct. 1, 2016, pp. 102–106. ISBN: 978-1-4503-3603-1 (cit. on p. 32).

[Wan17] WANG, Hongsong and WANG, Liang: "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks". In: *Conference on Computer Vision and Pa Ern Recognition (CVPR).* 2017 (cit. on pp. 31, 32, 114, 115, 134).

[Wan21a] WANG, Jinbao; TAN, Shujie; ZHEN, Xiantong; XU, Shuo; ZHENG, Feng; HE, Zhenyu and SHAO, Ling: "Deep 3D Human Pose Estimation: A Review". In: *Computer Vision and Image Understanding* 210 (Sept. 1, 2021), p. 103225. ISSN: 1077-3142 (cit. on pp. 14, 21, 22).

[Wan21b] WANG, Jing; WU, ZhongCheng; LI, Fang and ZHANG, Jun: "A Data Augmentation Approach to Distracted Driving Detection". In: *Future Internet* 13.1 (1 Jan. 2021), p. 1. ISSN: 1999-5903 (cit. on p. 39).

[Wan21c] WANG, Shun; ZHOU, Fang; CHEN, Song-Lu and YANG, Chun: "Recurrent Graph Convolutional Network for Skeleton-Based Abnormal Driving Behavior Recognition". In: *Pattern Recognition. ICPR International Workshops and Challenges.* Ed. by DEL BIMBO, Alberto; CUCCHIARA, Rita; SCLAROFF, Stan; FARINELLA, Giovanni Maria; MEI, Tao; BERTINI, Marco; ESCALANTE, Hugo Jair and VEZZANI, Roberto. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 551–565. ISBN: 978-3-030-68790-8 (cit. on p. 42).

[Wei16] WEI, Shih-En; RAMAKRISHNA, Varun; KANADE, Takeo and SHEIKH, Yaser: "Convolutional Pose Machines". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 4724–4732 (cit. on p. 16).

[Wei22] WEI, Jing; TAG, Benjamin; TRIPPAS, Johanne R; DINGLER, Tilman and KOSTAKOS, Vassilis: "What Could Possibly Go Wrong When Interacting with Proactive Smart Speakers? A Case Study Using an ESM Application". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems.* CHI '22. New York, NY, USA: Association for Computing Machinery, Apr. 29, 2022, pp. 1–15. ISBN: 978-1-4503-9157-3 (cit. on p. 9).

[Wey19] Weyers, Patrick; Schiebener, David and Kummert, Anton: “Action and Object Interaction Recognition for Driver Activity Classification”. In: *International Conference on Intelligent Transportation Systems (ITSC).* Oct. 2019, pp. 4336–4341 (cit. on p. 43).

[Wha21] Wharton, Zachary; Behera, Ardhendu; Liu, Yonghuai and Bessis, Nik: “Coarse Temporal Attention Network (CTA-Net) for Driver’s Activity Recognition”. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021, pp. 1279–1289 (cit. on pp. 40, 134).

[Wil20] Wilson, Kyle M.; Yang, Shiyan; Roady, Trey; Kuo, Jonny and Lenné, Michael G: “Driver Trust & Mode Confusion in an On-Road Study of Level-2 Automated Vehicle Technology”. In: *Safety Science* 130 (Oct. 1, 2020), p. 104845. issn: 0925-7535 (cit. on p. 3).

[Wor15] World Health Organization: Global Status Report on Road Safety 2015. Geneva, Switzerland, 2015 (cit. on p. 2).

[Wör21] Wörle, Johanna; Metz, Barbara and Baumann, Martin: “Sleep Inertia in Automated Driving: Post-sleep Take-over and Driving Performance”. In: *Accident Analysis & Prevention* 150 (Feb. 1, 2021), p. 105918. issn: 0001-4575 (cit. on p. 3).

[Wu21] Wu, Mingyan; Zhang, Xi; Shen, Linlin and Yu, Hang: “Pose-Aware Multi-feature Fusion Network for Driver Distraction Recognition”. In: *International Conference on Pattern Recognition (ICPR).* Jan. 2021, pp. 1228–1235 (cit. on p. 42).

[Xia17] Xiaohan Nie, Bruce; Wei, Ping and Zhu, Song-Chun: “Monocular 3D Human Pose Estimation by Predicting Depth on Joints”. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 3447–3455 (cit. on p. 22).

[Xin18] Xing, Yang; Lv, Chen; Zhang, Zhaozhong; Wang, Huaji; Na, Xiaoxiang; Cao, Dongpu; Velenis, Efstathios and Wang, Fei-Yue: “Identification and Analysis of Driver Postures for In-Vehicle Driving Activities and Secondary Tasks Recognition”. In: *IEEE Transactions on Computational Social Systems* 5.1 (Mar. 2018), pp. 95–108. issn: 2329-924X (cit. on pp. 26, 43).

[Xin19] Xing, Y.; Lv, C.; Wang, H.; Cao, D.; Velenis, E. and Wang, F.: “Driver Activity Recognition for Intelligent Vehicles: A Deep Learning Approach”. In: *IEEE Transactions on Vehicular Technology* 68.6 (June 2019), pp. 5379–5390. issn: 1939-9359 (cit. on p. 39).

[Xu14] Xu, Lijie and Fujimura, Kikuo: “Real-Time Driver Activity Recognition with Random Forests”. In: *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications.* AutomotiveUI ’14. New York, NY, USA: Association for Computing Machinery, Sept. 17, 2014, pp. 1–8. isbn: 978-1-4503-3212-5 (cit. on p. 38).

[Yam16] Yamada, Takahiro; Irie, Hidetsugu and Sakai, Shuichi: “High-Accuracy Joint Position Estimation and Posture Detection System for Driving”. In: *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services.* MOBIQUITOUS 2016. New York, NY, USA: Association for Computing Machinery, Nov. 28, 2016, pp. 219–224. isbn: 978-1-4503-4759-4 (cit. on p. 27).

[Yan14] Yan, Chao; Coenen, Frans and Zhang, Bailing: “Driving Posture Recognition by Joint Application of Motion History Image and Pyramid Histogram of Oriented Gradients”. In: *International Journal of Vehicular Technology* 2014 (2014), e719413. issn: 1687-5702 (cit. on pp. 36, 37, 40).

[Yan16a] Yan, Chao; Coenen, Frans; Yue, Yong; Yang, Xiaosong and Zhang, Bailing: “Video-Based Classification of Driving Behavior Using a Hierarchical Classification System with Multiple Features”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 30.05 (June 2016), p. 1650010. issn: 0218-0014 (cit. on pp. 35, 40).

[Yan16b] Yan, Chao; Coenen, Frans and Zhang, Bailing: “Driving Posture Recognition by Convolutional Neural Networks”. In: *IET Computer Vision* 10.2 (2016), pp. 103–114. issn: 1751-9640 (cit. on p. 37).

[Yan16c] Yan, Shiyang; Teng, Yuxuan; Smith, Jeremy S. and Zhang, Bailing: “Driver Behavior Recognition Based on Deep Convolutional Neural Networks”. In: *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).* Aug. 2016, pp. 636–641 (cit. on pp. 35, 39).

[Yan18] Yan, Sijie; Xiong, Yuanjun and Lin, Dahua: “Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition”. In: *Thirty-Second AAAI Conference on Artificial Intelligence.* Apr. 27, 2018 (cit. on pp. 33, 34, 42, 120, 121, 134).

[Yan19] Yang, Zhengyuan; Li, Yuncheng; Yang, Jianchao and Luo, Jiebo: “Action Recognition With Spatio–Temporal Visual Attention on Skeleton Image Sequences”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.8 (Aug. 2019), pp. 2405–2415. issn: 1558-2205 (cit. on p. 32).

[Yan21] Yang, Lichao; Yang, Ting-Yu; Liu, Haochen; Shan, Xiaocai; Brighton, James; Skrypchuk, Lee; Mouzakitis, Alexandros and Zhao, Yifan: "A Refined Non-Driving Activity Classification Using a Two-Stream Convolutional Neural Network". In: *IEEE Sensors Journal* 21.14 (July 2021), pp. 15574–15583. issn: 1558-1748 (cit. on p. 40).

[Yao20] Yao, Zhijie; Liu, Yazhou; Ji, Zexuan; Sun, Quansen; Lasang, Pongsak and Shen, Shengmei: "3D Driver Pose Estimation Based on Joint 2D–3D Network". In: *IET Computer Vision* 14.3 (2020), pp. 84–91. issn: 1751-9640 (cit. on p. 28).

[Ye20] Ye, Fanfan; Pu, Shiliang; Zhong, Qiaoyong; Li, Chao; Xie, Di and Tang, Huiming: "Dynamic GCN: Context-enriched Topology Learning for Skeleton-based Action Recognition". In: *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 12, 2020, pp. 55–63. isbn: 978-1-4503-7988-5 (cit. on p. 34).

[Yub15] Yub Jung, Ho; Lee, Soochahn; Seok Heo, Yong and Dong Yun, Il: "Random Tree Walk Toward Instantaneous 3D Human Pose Estimation". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 2467–2474 (cit. on p. 19).

[Yue19] Yuen, Kevan and Trivedi, Mohan M.: "Looking at Hands in Autonomous Vehicles: A ConvNet Approach Using Part Affinity Fields". In: *IEEE Transactions on Intelligent Vehicles* (2019), pp. 1–1. issn: 2379-8858 (cit. on pp. 23–25, 27).

[Zha12a] Zhao, C.H.; Zhang, B.L.; He, J. and Lian, J.: "Recognition of Driving Postures by Contourlet Transform and Random Forests". In: *IET Intelligent Transport Systems* 6.2 (June 2012), pp. 161–168. issn: 1751-9578 (cit. on pp. 35, 37, 39).

[Zha12b] Zhao, Chihang; Gao, Yongsheng; He, Jie and Lian, Jie: "Recognition of Driving Postures by Multiwavelet Transform and Multilayer Perceptron Classifier". In: *Engineering Applications of Artificial Intelligence* 25.8 (Dec. 1, 2012), pp. 1677–1686. issn: 0952-1976 (cit. on p. 39).

[Zha13] Zhao, Chihang H.; Zhang, Bailing L.; Zhang, Xiaozheng Z.; Zhao, Sanqiang Q. and Li, Hanxi X.: "Recognition of Driving Postures by Combined Features and Random Subspace Ensemble of Multilayer Perceptron Classifiers". In: *Neural Computing and Applications* 22.1 (May 1, 2013), pp. 175–184. issn: 1433-3058 (cit. on p. 39).

[Zha17a] Zhang, Pengfei; Lan, Cuiling; Xing, Junliang; Zeng, Wenjun; Xue, Jianru and Zheng, Nanning: “View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data”. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2117–2126 (cit. on p. 31).

[Zha17b] Zhang, Songyang; Liu, Xiaoming and Xiao, Jun: “On Geometric Features for Skeleton-Based Action Recognition Using Multilayer LSTM Networks”. In: *Winter Conference on Applications of Computer Vision (WACV).* Mar. 2017, pp. 148–157 (cit. on p. 31).

[Zha18] Zhao, Mingming; Beurier, Georges; Wang, Hongyan and Wang, Xuguang: In Vehicle Diver Postural Monitoring Using a Depth Camera Kinect. SAE Technical Paper 2018-01-0505. Warrendale, PA: SAE International, Apr. 3, 2018 (cit. on p. 27).

[Zha19a] Zhang, C.; Wu, X.; Zheng, X. and Yu, S.: “Driver Drowsiness Detection Using Multi-Channel Second Order Blind Identifications”. In: *IEEE Access* 7 (2019), pp. 11829–11843. issn: 2169-3536 (cit. on p. 6).

[Zha19b] Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J. and Zheng, N.: “View Adaptive Neural Networks for High Performance Skeleton-based Human Action Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), pp. 1–1. issn: 0162-8828 (cit. on p. 31).

[Zha19c] Zhao, Long; Peng, Xi; Tian, Yu; Kapadia, Mubbasir and Metaxas, Dimitris N.: “Semantic Graph Convolutional Networks for 3D Human Pose Regression”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, pp. 3425–3435 (cit. on p. 22).

[Zha21] Zhao, Lei; Yang, Fei; Bu, Lingguo; Han, Su; Zhang, Guoxin and Luo, Ying: “Driver Behavior Detection via Adaptive Spatial Attention Mechanism”. In: *Advanced Engineering Informatics* 48 (Apr. 1, 2021), p. 101280. issn: 1474-0346 (cit. on p. 39).

[Zim18] Zimmermann, Christian; Welschehold, Tim; Dornhege, Christian; Burgard, Wolfram and Brox, Thomas: “3D Human Pose Estimation in RGBD Images for Robotic Task Learning”. In: *International Conference on Robotics and Automation (ICRA).* May 2018, pp. 1986–1992 (cit. on p. 20).

Own publications

Authors marked with an asterisk (*) contributed equally to the publication.

[1] Martin, Manuel; Li, Kangxiong; Voit, Michael; Melcher, Vivien; Widlroither, Harald; Diederichs, Frederik and Stiefelhagen, Rainer: “Klassifikation von Fahrerzuständen Und Nebentätigkeiten Über Körperposen Bei Automatisierter Fahrt”. In: *32. VDI/VW Gemeinschaftstagung ”Fahrerassistenzsysteme Und Automatisiertes Fahren”*. 2016.

[2] Martin, Manuel; Stuehmer, Stephan; Voit, Michael and Stiefelhagen, Rainer: “Real Time Driver Body Pose Estimation for Novel Assistance Systems”. In: *International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 1–7.

[3] *Ludwig, Julian; *Martin, Manuel; Horne, Matthias; Flad, Michael; Voit, Michael; Stiefelhagen, Rainer and Hohmann, Sören: “Driver Observation and Shared Vehicle Control: Supporting the Driver on the Way Back into the Control Loop”. In: *at - Automatisierungstechnik* 66.2 (2018), pp. 146–159. issn: 0178-2312.

[4] Martin, Manuel; Popp, Johannes; Anneken, Mathias; Voit, Michael and Stiefelhagen, Rainer: “Body Pose and Context Information for Driver Secondary Task Detection”. In: *Intelligent Vehicles Symposium (IV)*. 2018, pp. 2015–2021.

[5] *Martin, Manuel; *Roitberg, Alina; Haurilet, Monica; Horne, Matthias; Reiss, Simon; Voit, Michael and Stiefelhagen, Rainer: “Drive&Act: A Multi-Modal Dataset for Fine-Grained Driver Behavior Recognition in Autonomous Vehicles”. In: *International Conference on Computer Vision (ICCV)*. 2019, pp. 2801–2810.

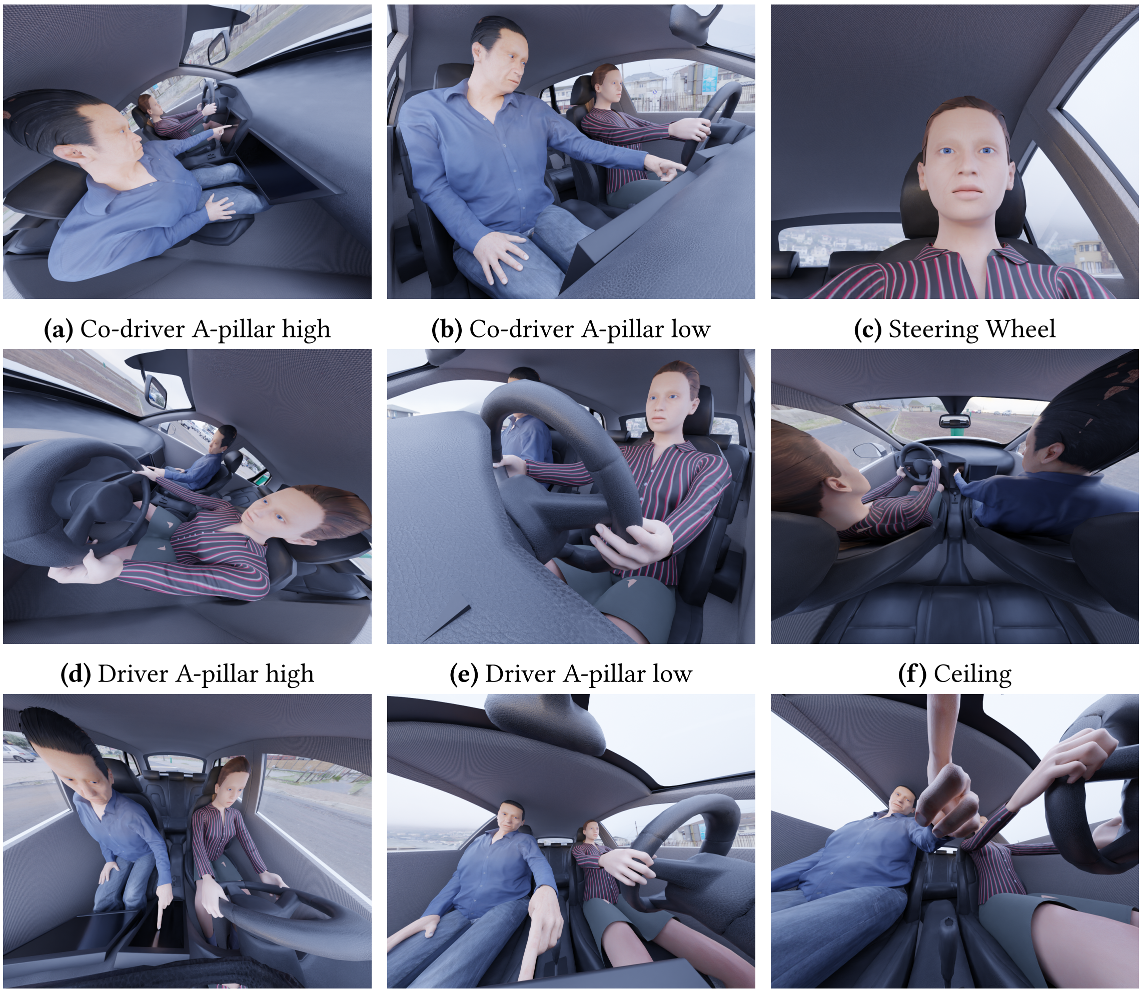
[6] Martin, Manuel; Ludwig, Julian; Roitberg, Alina; Flad, Michael; Voit, Michael; Hohmann, Sören and Stiefelhagen, Rainer: “Innenraumbeobachtung Für Die Kooperative Übergabe Zwischen Hochautomatisierten Fahrzeugen Und Fahrer”. In: *Der (Mit-)Fahrer im 21. Jahrhundert? 10. VDI Fachtagung Mensch-Maschine-Mobilität.* VDI-Berichte (2019), pp. 67–78. issn: 0083-5560.

[7] Flad, Michael; Karg, Philipp; Roitberg, Alina; Martin, Manuel et al.: "Personalisation and Control Transition Between Automation and Driver in Highly Automated Cars". In: *Smart Automotive Mobility: Reliable Technology for the Mobile Human.* Human–Computer Interaction Series. Cham: Springer International Publishing, 2020, pp. 1–70. isbn: 978-3-030-45131-8.

[8] Martin, Manuel; Voit, Michael and Stiefelhagen, Rainer: "Dynamic Interaction Graphs for Driver Activity Recognition". In: *International Conference on Intelligent Transportation Systems (ITSC).* 2020.

[9] Martin, Manuel; Voit, Michael and Stiefelhagen, Rainer: "An Evaluation of Different Methods for 3D-Driver-Body-Pose Estimation". In: *International Conference on Intelligent Transportation Systems (ITSC).* 2021.

A Simulated camera views



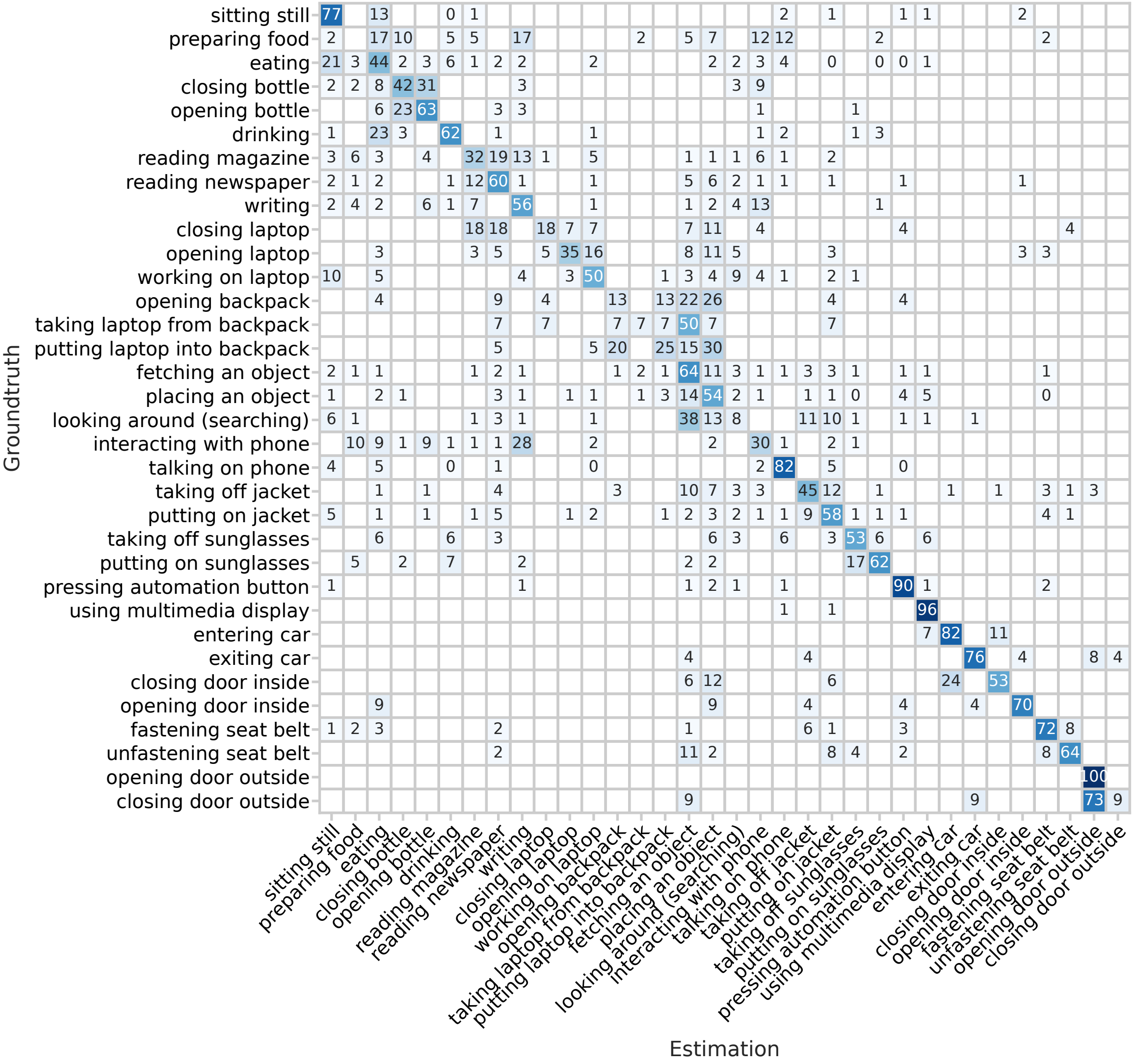
(a) Co-driver A-pillar high **(b)** Co-driver A-pillar low **(c)** Steering Wheel

(d) Driver A-pillar high **(e)** Driver A-pillar low **(f)** Ceiling

(g) Interior mirror **(h)** Dashboard center **(i)** Infotainment (low)

Figure A.1: Simulated camera views for sensor position evaluation.

B Interaction Graphs: Confusion Matrices for Drive&Act Using Different Input Modalities.



	sitting still	preparing food	eating	closing bottle	opening bottle	drinking	reading magazine	reading newspaper	writing	closing laptop	opening laptop	working on laptop	opening backpack	taking laptop from backpack	putting laptop into backpack	fetching an object	placing an object	looking around (searching)	interacting with phone	talking on phone	taking off jacket	putting on jacket	taking off sunglasses	putting on sunglasses	pressing automation button	using multimedia display	entering car	exiting car	closing door inside	opening door inside	fastening seat belt	unfastening seat belt	opening door outside	closing door outside
sitting still	77		13			0	1													2		1			1	1				2				
preparing food	2		17	10		5	5		17					2		5	7		12	12				2							2			
eating	21	3	44	2	3	6	1	2	2			2					2	2	3	4		0		0	0	1								
closing bottle	2	2	8	42	31				3									3	9															
opening bottle			6	23	63			3	3										1				1											
drinking	1		23	3		62		1				1							1	2			1	3										
reading magazine	3	6	3		4		32	19	13	1		5				1	1	1	6	1		2												
reading newspaper	2	1	2			1	12	60	1			1				5	6	2	1	1		1			1					1				
writing	2	4	2		6	1	7		56			1				1	2	4	13					1										
closing laptop							18	18		18	7	7				7	11		4						4							4		
opening laptop			3				3	5		5	35	16				8	11	5				3								3	3			
working on laptop	10		5						4		3	50			1	3	4	9	4	1		2	1											
opening backpack			4					9		4			13		13	22	26					4			4									
taking laptop from backpack								7		7			7	7	7	50	7					7												
putting laptop into backpack								5				5	20		25	15	30																	
fetching an object	2	1	1				1	2	1				1	2	1	64	11	3	1	1	3	3	1		1	1					1			
placing an object	1		2	1				3	1		1	1		1	3	14	54	2	1		1	1	0		4	5					0			
looking around (searching)	6	1					1	3	1			1				38	13	8			11	10	1		1	1		1						
interacting with phone		10	9	1	9	1	1	1	28			2					2		30	1		2	1											
talking on phone	4		5			0		1				0							2	82		5			0									
taking off jacket			1		1			4					3			10	7	3	3		45	12		1			1		1		3	1	3	
putting on jacket	5		1		1		1	5			1	2			1	2	3	2	1	1	9	58	1	1	1						4	1		
taking off sunglasses			6			6		3									6	3		6		3	53	6		6								
putting on sunglasses		5		2		7			2							2	2						17	62										
pressing automation button	1								1							1	2	1		1					90	1					2			
using multimedia display																				1		1				96								
entering car																										7	82		11					
exiting car																4					4							76		4			8	4
closing door inside																6	12					6					24		53					
opening door inside			9														9				4				4			4		70				
fastening seat belt	1	2	3					2								1					6	1			3						72	8		
unfastening seat belt								2								11	2					8	4		2						8	64		
opening door outside																																	100	
closing door outside																9												9					73	9

Figure B.1: Confusion matrix of the Dynamic Interaction Graph method on the fine-grained activity annotations of Drive&Act using only the 3D driver body pose as input.

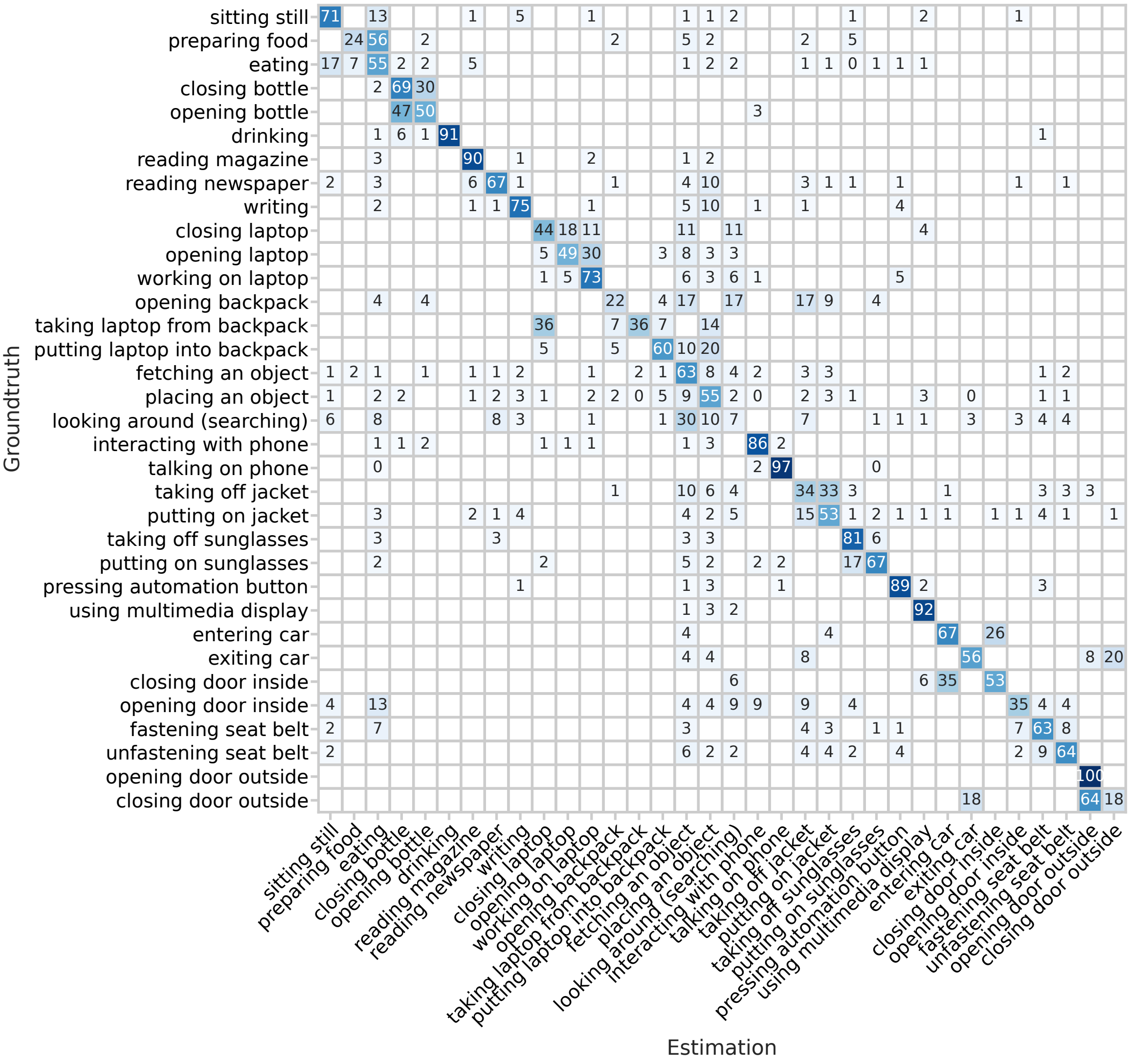


Figure B.2: Confusion matrix of the Dynamic Interaction Graph method on the fine-grained activity annotations of Drive&Act using all three input modalities (3D driver body pose, interior elements, objects).