Technische Universität München
TUM School of Computation, Information and Technology

Enabling Teleoperated Driving in Everyday's Traffic Scenarios

Stefan Neumeier

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr. Alexander Pretschner
Prüfer*innen der Dissertation:	1. Prof. Dr.-Ing. Jörg Ott
	2. Prof. Dr. rer. nat. Christian Facchi
	3. Prof. Dr.-Ing. Lars Wolf

Die Dissertation wurde am 11.01.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 30.05.2023 angenommen.

Declaration

I hereby declare that this thesis is my own work, that I have not presented it elsewhere for examination purposes and that I have not used any sources or aids other than those stated. I have marked verbatim and indirect quotations as such.

Parts of this work have been published as international conference papers or journal articles.

Peer-reviewed core publications based on their date are: [**VII**, **VIII**, **III**, **V**, **VI**, **I**]
Publications with peer-review process, but not considered for this thesis are: [**II**]
Publications without peer-review process are: [**IV**]
Further publications as co-author are: [**XII**, **XI**, **IX**, **X**]

Egweil, June 20, 2023

Stefan Neumeier

Abstract

Teleoperated driving, in which a human driver controls a vehicle remotely, may be a promising approach on the way to autonomous driving. In order to meet the special requirements of highly mobile vehicles, communication between the driver and the vehicle must take place via cellular networks. However, cellular networks are subject to potential fluctuations in their availability and performance, which can have a negative impact on the remote control of vehicles. In this thesis, we address this network issue, examine its implications and present possible solutions.

In the course of this thesis, we address the following five research questions: (1) *Is teleoperated driving feasible with contemporary cellular networks?*, (2) *Is it possible to achieve a safe remote control of a vehicle on typical roads despite latency using basic algorithms?*, (3) *What is the influence of variable and fixed latency on teleoperated driving performance and subjective assessment?*, (4) *How far can video streams be compressed to still allow for safe teleoperated driving?* and (5) *How can the required bandwidth for uplink video streams be decreased?*.

We answer the first research question with the help of contemporary cellular measurements that we conducted in Germany while driving cars. Based on these measurements, we were able to derive initial results for the application of teleoperated driving. These results show that teleoperated driving is possible in principle. However, the high variance of the network parameters makes it difficult to use such a system at all times.

In the context of the second research question, we present an algorithm that suggests speed reductions depending on the latency of the cellular network, thus compensating for the existing latency. We also show that this approach is promising for teleoperated driving.

To answer the third research question, we conducted a user study in which participants had to drive different routes with different latencies in a driving simulator. We measured the objective driving performance as well as the perceived workload. Our results show that driving performance decreases while perceived workload increases by adding latency.

In order to investigate more than just latency, we address the issue of bandwidth in the context of video streams as part of the fourth research question. For this purpose, we conducted a user study in which the participants had to rate different video clips of real driving situations, which were at different levels of displayed quality. Results show that the driving situation has a strong influence on the rated quality.

Finally, in the context of the fifth research question, we test an approach in which we divide the video stream into important and less important areas in order to reduce the required bandwidth. A filter is applied to the less important areas. Our results show that the bandwidth can be further reduced by this approach. The conducted user study revealed that the displayed quality was sufficient for teleoperated driving.

Kurzfassung

Teleoperiertes Fahren, bei welchem ein menschlicher Fahrer ein Fahrzeug aus der Ferne steuert, kann ein vielversprechender Ansatz auf dem Weg zum autonomen Fahren sein. Um dabei die speziellen Anforderungen hochmobiler Fahrzeuge abdecken zu können, muss die Kommunikation zwischen Fahrer und Fahrzeug über Mobilfunknetze erfolgen. Mobilfunknetze unterliegen aber potenziellen Schwankungen in ihrer Verfügbarkeit und Performance, welche einen negativen Einfluss auf die Fernsteuerung von Fahrzeugen haben kann. Im Rahmen diese Arbeit nehmen wir uns dieser Netzwerkproblematik an, untersuchen die Auswirkungen und präsentieren Lösungsansätze.

Im Zuge dessen bearbeiten wir die folgenden fünf Forschungsfragen: (1) *Ist teleoperiertes Fahren mit aktuellen Mobilfunknetzen möglich?*, (2) *Ist es möglich, mit einfachen Algorithmen das sichere Fernsteuern eines Fahrzeugs auf normalen Straßen trotz Latenz zu erreichen?*, (3) *Welchen Einfluss haben variable und feste Latenzzeiten auf das teleoperierte Fahren und die subjektive Einschätzung?*, (4) *Wie weit können Viedeoströme komprimiert werden und gleichzeitig noch ein sicheres teleoperiertes Fahren ermöglichen?*, (5) *Wie kann die erforderliche Bandbreite für Uplink-Videoströme verringert werden?*.

Die erste Forschungsfrage beantworten wir mit Hilfe von Mobilfunkmessungen, die wir in Deutschland durchgeführt haben, während wir mit Autos unterwegs waren. Auf Basis dieser Messungen konnten wir erste Ergebnisse für die Anwendung von teleoperiertem Fahren ableiten. Diese Ergebnisse zeigen, dass teleoperiertes Fahren prinzipiell möglich ist. Die hohe Varianz der Netzparameter macht es jeodch schwierig, solch ein System durchgängig zu nutzen.

Im Rahmen der zweiten Forschungsfrage stellen wir einen Algorithmus vor, der abhängig von der Latenz des Mobilfunknetzes Vorschläge zur Verrigerung der Geschwindigkeit macht und somit die vorhandene Latenz ausgleicht. Wir zeigen zudem, dass dieser Ansatz vielver-

sprechend für teleoperiertes Fahren ist.

Zur Beantwortung der dritten Forschungsfragen haben wir eine Nutzerstudie durchgeführt, bei der Teilnehmer*innen verschiedene Strecken mit unterschiedlichen Latenzen in einem Fahrsimulator fahren mussten. Dabei haben wir die objektive Fahrleistung sowie die subjektiv wahrgenommene Belastung gemessen. Unsere Ergenisse zeigen, dass sich sowohl die Fahrleistung als auch die wahrgenomme Belastung durch das hinzufügen von Latenz verschlechtern.

Um nicht nur Latenz zu untersuchen, haben wir uns im Rahmen der vierten Forschungsfrage mit dem Thema der Bandbreite im Kontext von Videoströmen auseinandergesetzt. Dazu haben wir eine Nutzerstudie durchgeführt, in welcher die Teilnehmer*innen verschiedene Videoclips von realen Fahrsituation unterschiedlicher Darstellungsqualität bewerten mussten. Die Ergebnisse zeigen, dass die jeweilige Fahrsituation einen starken Einfluss auf die bewertete Darstellungsqualität haben.

Abschließend haben wir im Kontext der fünften Forschungsfrage einen Ansatz getestet, bei welchem wir den Videostrom in wichtige und weniger wichtige Bereiche aufteilen um die benötigte Bandbreite zu reduzieren. Die weniger wichtigen Bereiche werden dabei mit einem Filter versehen. Unsere Ergebnisse zeigen, dass die Bandbreite mit diesem Ansatz weiter gesenkt werden kann und dass die im Rahmen einer Nutzerstudie ermittelte Darstellungsqualität ausreichend ist für teleoperiertes Fahren.

Acknowledgment

The work presented in this thesis was carried out at the Chair of Connected Mobility at Technische Universität München (TUM) and the C-ECOS at Technische Hochschule Ingolstadt (THI) being part of the BayWISS Verbundkolleg "Mobilität und Verkehr". I would like to thank my supervisor at TUM, Prof. Dr.-Ing. Jörg Ott, for supporting me during all phases of my thesis and for also supporting me on all publications we worked on together. He also provided me the chance to be with his chair, proposed interesting research directions and gave me all support I could ask for. This opened a lot of opportunities for my research and I am very grateful for this. In addition, I would like to thank Prof. Dr. rer. nat. Christian Facchi, my supervisor at THI. He was always there when I needed help and provided me with great support during the thesis and on all the papers. Special thanks also goes to Dr. Vaibhav Bajpai, who supported me during my time at TUM. Furthermore, I would also like to thank Prof. Dr.-Ing. Lars Wolf for being third examiner in my thesis and Prof. Pretschner for acting as chair. In general I would like to thank all the colleagues that I have been working with. It was a pleasure to work with you and have inspiring discussion on research topics. Thanks also to all other colleagues with whom I was allowed to write exciting papers: Andreas Riener, Anna-Katharina Frison, Armin Becher, Christopher Corbett, Clemens Dannheim, Ermias Andargie Walelgne, Marion Neumeier, Michael Höpp, Nicolas Gay, Philipp Wintersberger and Simon Stapf. I really appreciated working with you and writing these papers. Thank you also to the team of the Elitenetzwerk Bayern, who provided me with the chance of becoming funded and supported by the Promotionsstipendium, which really helped me in carrying out my research. Finally, a big thank you goes to my family, my girlfriend and my friends. Without you and your support, I would not have had the chance to work on this thesis and I would not be who I am now.

Stefan Neumeier, Egweil, June 20, 2023

Contents

List of Figures

List of Tables

Acronyms

ABS Antilock Braking System. xv, 3

ACC Adaptive Cruise Control. xv, 3

ADTF Automotive Data and Time-Triggered Framework. xv, 28

BBR Bottleneck Bandwidth and Round-trip propagation time. xv, 23

BF Blur-Full. xv, 76–78, 81

BLOS Beyond Line of Sight. xv, 9

COCO Common Objects in Context. xv, 78

CRF Constant Rate Factor. xv, xxxiii, 28, 71, 90

ERTRAC European Road Transport Research Advisory Council. xi, xv, 4, 5

ESP Electronic Stability Program. xv, 3

FOV Field of View. xv, 76, 78, 81

GBF Gray Blur-Full. xv, 76–78, 81

GDPR General Data Protection Regulation. xv, 42

GPL GNU General Public License. xv, 28, 44, 55

HiL Hardware-in-the-Loop. xv, 28

HMD Head-Mounted Display. xv, 24, 57

HMI Human-Machine Interface. xv, 25

HSPA High Speed Packet Access. xv, 50

ICMP Internet Control Message Protocol. xii, xv, 23, 47–51, 53

LiDAR Light Detection and Ranging. ii, xv, 17, 24, 27

LKAS Lane Keeping Assist Systems. xv, 3

LOS Line of Sight. xv, 9

LTE Long Term Evolution. xv, xxiii, 21–23, 47, 52, 54

MAC Message Authentiation Code. xv, 41

MADRaS Multi-Agent DRiving Simulator. xv, 29

mAP mean Average Precision. xv, 78

MITM Man-in-the-Middle. xv, 41

MLP Mean Lateral Position. xv, 64, 66

MOS Mean Opinion Score. xv, lii, 73, 81–83, 90, 91

MS-SSIM Multiscale SSIM. xv, 72

MTOW Maximum Takeoff Weight. xv, 8

NASA-TLX NASA Task Load Index. ix, xv, 61, 64, 67, 68

OEM Original Equipment Manufacturer. xv, 39

OpenDS Open Source Driving Simulation. xv, 28

OpenROUTS3D Open Realtime OSM- and Unity-based Traffic Simulator 3D. xii, xv, 31, 44, 46, 55–61, 85

OSM Openstreetmap. xv, 35, 56, 60

PSNR Peak signal-to-noise ratio. xv

QoE Quality of Experience. xv, 26

QoS Quality of Service. xv, 16–18, 45, 46

ROS Robot Operating System. xv, 28, 29

ROUV Remotely Operated Underwater Vehicle. xi, xv, 9, 10

RTT Round Trip Time. xv, 21–23, 48, 53, 88

SDLP Standard Deviation of Lateral Position. xv, 64, 66

SDN Software Defined Networking. xv, 23

SiL Software-in-the-Loop. xv, 28

SSIM Structural Similarity Index Measure. xv

SUMO Simulation of Urban MObility. i, xv, 55–57, 59

TCP Transmission Control Protocol. xv, 17, 22, 23, 53

TraCI Traffic Control Interface. xv, 55, 57

TTL Time to Live. xv, 41

UAS Unmanned Aerial System. xv, 8

UAV Unmanned Aerial Vehicle. xi, xv, 8, 9

UDP User Datagram Protocol. xv, 17, 22, 27, 41, 48, 49, 53, 54, 58

UGV Unmanned Ground Vehicle. xv, 10

UI User Interface. xv, 60

VMAF Video Multi-Method Assessment Fusion. xv, xxxiii, xxxiv, 71, 72, 74, 75, 86, 90

VR Virtual Reality. xv, 25

WLAN IEEE 802.11 Wireless Local Area Network. xv, 22, 23

1 Introduction

New technologies can help to improve various areas of life and work. Road-legal vehicles, i. e., all typical vehicles on public roads, are no exception. Especially comfort- and safety-related features increased drastically (e. g., AirBags [133]) during the last decades. With the development of advanced driver assistance systems [20] and additional systems for passenger protection, safety is improved, leading to a decrease of fatalities and severe injuries [104]. Considering the 2019 mid-class limousine BMW series 3 model as an example, emergency brake systems and an active bonnet are already available in the basic configuration [52]. Overall, the development of new vehicles aims for higher levels of automation, as can be seen by Mercedes allowing autonomous driving in certain highway situations already [123]. These autonomous vehicles are assumed being key technology to further reduce the number of traffic incidents and fatalities. Furthermore, they help to improve driving comfort, e. g., by minimizing the stress level of the driver. Waymo, as a leading company for developing highly autonomous vehicles, is offering a real-world chauffeur service for "all customers of its ride-hailing service in Phoenix" [79], enabling more than thousand potential passengers to use autonomous vehicles [79]. Nevertheless, such systems can only be used in well-known environments under certain limited environmental conditions and are therefore still not available generally. Considering the well-defined SAE levels of automation [148], that range from *Level 0* as *No Automation* to *Level 5* as *Full Automation*, for vehicles available in 2021 for purchase, it turns out that they mainly operate on *Level 2*. In *Level 2* drivers are assisted in lateral and longitudinal control. Vehicles that provide *Level 3* functionality, where drivers are allowed to put their attention away from the street under certain conditions, are announced by multiple companies, but are mostly not yet available to the broad public [30, 102] and only included in first Mercedes vehicles [123]. Besides overcoming a number of technical challenges, a new way of testing and regulatory approval is required to enable the broad usage of non-test autonomous vehicles in everyday traffic situations [114]. For example, with the *Act on Autonomous Driving* [55] regulatory rules for *Level 4* for non-test autonomous vehicles are present in Germany [55]. Even the currently most advanced systems are not flawless. The Waymo fleet, for example, was involved in "18 crashes and 29 near-miss collisions during 2019 and the first nine months of 2020" [78]. Although nobody was injured and most of the crashes were the fault of other traffic participants [78], it will take additional time to adjust the autonomous driving functionality to further cities or even other countries. Autonomous driving vehicles of other manufacturers are also not flawless and involved in crashes [54, 120, 81]. As an example, a fatal crash was caused by an autonomous test vehicle of Uber in 2018, where a street-crossing pedestrian was struck by an autonomously acting vehicle.[108].

Although fully autonomous vehicles (SAE *Level 5*) are the ultimate goal, there will be a period where vehicles are not fully autonomous, as there will be situations that cannot be solved by them, e. g., driving on field-ways because of a blocked street or complex road side works. As such, further supporting technology is required to allow a fruitful and safe introduction of autonomous driving. A promising technology claimed to foster the breakthrough and development of autonomous systems [63] is teleoperated driving, also known as remote driving.

Taxonomy of Autonomous Vehicles

In the norm SAE J3016 [148] the autonomous driving features are split up into six different levels, based on the responsibility for

- "Execution of Steering and Acceleration/Deceleration", the control of lateral and longitudinal steering (human and/or system).
- "Monitoring of Driving Environment", the monitoring of the environment (human or system), which defines whether the human driver may focus on other tasks than watching the traffic.
- "Fallback Performance of Dynamic Driving Task", the fallback if the system fails (human or system), defining whether the human driver needs to be able to take over control if the system cannot handle a situation anymore.

Table 1.1: The six different levels of automation based on the classification of the norm SAE J3016 [148].

Level	Name	Lon/Lat steering	Monitoring	Fallback
0	No Automation	Driver	Driver	None
1	Driver Assistance	Driver and System	Driver	None
2	Partial Automation	System	Driver	Driver
3	Conditional Automation	System	System	Driver
4	High Automation	System	System	System
5	Full Automation	System	System	System

- "System Capability (Driving Modes)", the driving modes that can be executed by the system.

These six levels reach from 0 to 5, representing *No Automation* up to *Full Automation*. An overview of the different levels together with their specific names and the assigned responsibilities is given in Table 1.1.

Although the name of ***Level 0***, *No Automation*, may suggest a vehicle without any supporting systems, this can be misleading. It only implies that there is no automation with respect to the responsibilities introduced previously. Thus, this means that the systems are just limited in their active support to the driver, e. g., Antilock Braking System (ABS). The driver still needs to control the vehicle, but in certain critical situations those systems support in controlling the vehicle. Within ***Level 1***, *Driver Assistance*, first automated features can be seen. Such a system is able to control either lateral or longitudinal steering, e. g., offering a lane centering system or an Adaptive Cruise Control (ACC). ***Level 2***, *Partial Automation*, enhances *Level 1* by both lateral and longitudinal steering being controlled by the system, e. g., an advanced ACC with lane keeping assistant, as shown by Mercedes in [122]. For all of these three levels, the person in the driver's seat of the vehicle is still responsible for monitoring the environment and permanently supervising the active systems, i. e., the driver needs to be able to control the vehicle if required at all times. [148]

Most of the cars built in 2020 are equipped with features that can be classified as *Level 1* or *Level 2*. However, in 2022 Mercedes Benz introduced the world's first cars equipped with the

Drive Pilot supporting *Level 3* on highways, that can be purchased by individuals [123, 80].

Starting from ***Level 3***, *Conditional Automation*, the system takes care of longitudinal and lateral steering, but is additionally responsible for monitoring the environment. Nevertheless, the capabilities of Level 3 systems are strongly limited to specific driving situations. Within this level, the driver still acts as a fallback and needs to react within as specific amount of time as soon as the system prompts him to do so.

By reaching ***Level 4***, *High Automation*, a large step is made. The system takes care of longitudinal and lateral steering and is also responsible for monitoring the environments at all times. This means that the person sitting in the driver seat will not be responsible for taking over anymore. However, the system will not operate in conditions or areas it is not explicitly suited for. At this level, vehicles no longer need a steering wheel and pedals and therefore may not enter areas they are not meant for.

The major difference between *Level 4* and ***Level 5***, *Full Automation*, is that *Level 5* systems can handle all situations a human driver would be able to handle. [148]

Besides the level of *Full Automation*, nowadays the most interesting point is the transition from *Level 2* to *Level 3* or directly to *Level 4*. Up to and including *Level 3* the driver must be always prepared to intervene. From *Level 3* onward the driver can (partially) rely on the system and is not responsible in the sense of driving; at least in the predefined conditions and areas the system is meant for. [148] Although current developments indicate a fast introduction of autonomous vehicles, it is still a long way to reach the point where those types of vehicles are widely deployed on streets. Based on the estimation of the European Road Transport Research Advisory Council (ERTRAC) Task Force *Connectivity and Automated Driving*, which is now a working group of the ERTRAC, vehicles with fully autonomous features will be available earliest in around 2028 [13], but this might still be very optimistic.

With the lack of full area and situation coverage by *Level 4* vehicles, it is important to have a fallback and supporting solution until *Level 5* vehicles are available. Since there may be a lack of suitable drivers and/or steering wheels and pedals, one solution may be to bring in the experience of a driver from remote, as is the case with teleoperated driving.

1.1 Teleoperated Driving

Teleoperated driving is the remote control of vehicles by a human operator in specific situations, that require human skills and knowledge. One such scenario is complex road-side works at which autonomous vehicles can easily fail. Especially in the early phases of *Level 4* vehicles, software and hardware failures [93] can happen. Teleoperated driving systems are under research and already developed by various start-ups such as *StarSky Robotics* [163], *Phantom Auto* [144], *Designated Driver* [46], *Fernride* [57], car manufacturers like *Nissan* [38], and telecommunication companies like *Ericsson* [51]. Another example is *Einride* [49], which is developing trucks that can drive autonomously, but have the option to be remotely controlled [77]. For testing driverless vehicles teleoperated driving is mandatory in California by law and multiple countries consider requiring the same to allow testing of autonomous vehicles [39].

1.1.1 Variants of Teleoperation

Various fields of application are at hand for using teleoperation in general. With the *DA VINCI* system [90], a widely used and well known system for remote surgery exists. This system is used for minimally invasive surgery as it translates the "surgeon's hand movements at the console in real time" [90]. Combined with an enhanced high resolution view, it allows for a safe and minimal surgery and therefore helps to protect against severe impacts of operations. [90]

In a domain more related to teleoperated driving, the remote control of vehicles, Mars rovers are probably one of the best known applications. NASA's Mars rover *Curiosity*, which can be seen in Figure 1.1, is one example of such rovers. Landed on Mars in 2012 and equipped with a set of different sensors, it is meant to analyze stones and other material on Mars to explore whether Mars is or was able to make life possible. It is about 3 m long, 2.7 m wide, 2.2 m high and has a weight of about 900 kg [131]. A complex system of three antennas is used for the communication between the ground station on earth and the rover on Mars, enabling bidirectional data transfer. One antenna is meant to communicate with the mars rover and sends data from it to earth utilizing the capabilities of Mars orbiters, e. g., NASA's Mars *Odyssey*. This allows for a longer line of sight with earth and as such provides higher data rates of up to 2 Mbps from rover to orbiter. In addition, this helps to save power on the

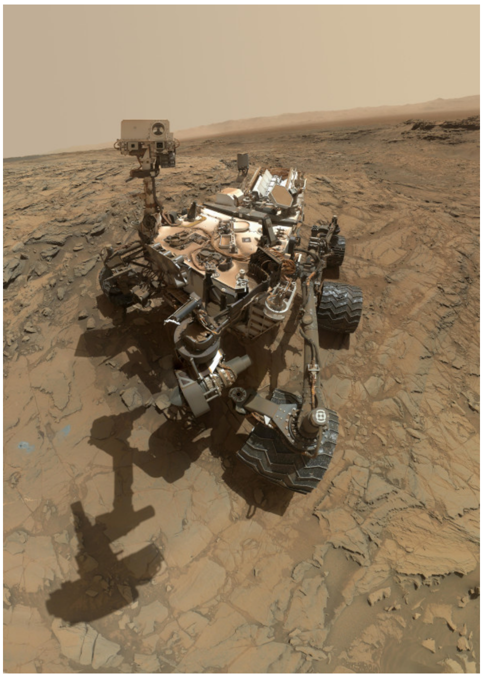


Figure 1.1: Mars Rover Curiosity with its self-portrait on Mars (Source: [132]).

rover as it does not need to send directly to the ground station on earth, but only to the much closer orbiter [130]. Another antenna is used for communicating directly with earth and is mainly meant for receiving mission commands. With a low bandwidth of about 160 bps to 800 bps and latencies of 5 – 20 minutes for signals between earth and Mars [129], the rover has to be able to execute received commands predominant autonomously, i. e., commands tend to be of a higher abstraction level [128, 160, 106].

Further systems that are known in the field of remote vehicles are Unmanned Aerial Vehicles (UAVs), which are often also referred to as *Unmanned Aerial Systems (UASs)* [35]. They can be defined as "a reusable aircraft designed to operate without an onboard pilot. It does not carry passengers and can be either remotely piloted or preprogrammed to fly autonomously." [162]

An example of an UAV is the *Heron-1* shown in Figure 1.2. Such systems can be classified based on their size, capabilities, operational conditions, etc. [34]. Usually the Maximum Takeoff Weight (MTOW) together with the ground impact risk is populated to classify *UAVs*, e. g., as done by [36] with six classes ranging from less than 1 kg (Micro) to more than 4,332 kg (Large). Another interesting classification arises from the level of autonomy an UAV can

Figure 1.2: Heron-1 UAV that can be used to support in disaster aid (DLR, CC-BY 3.0; Source: [41]).

provide. Following the classification of [88], this can be:

- **Remotely Piloted**, where the UAV is directly controlled by an operator.
- **Remotely operated (semiautonomous)**, where the UAV is provided with high level commands (e. g., waypoints) and performs them on its own. The operator is required to monitor the performance and make decisions.
- With **fully autonomous** UAVs, the operator is only required to specify the goal. The UAV is able to find a plan on how to achieve the goal and will execute this plan accordingly. Additionally, the UAV can react to unforeseen events.

With different levels of autonomy, the required bandwidth to ensure a stable communication line with the UAVs varies. While the bandwidth requirements are comparably high if the system is remotely piloted, the bandwidth requirements are lower if the UAV flies fully autonomous. This also applies to the reliability and latency of the network connection. With two different operational modes within the Line of Sight (LOS) and Beyond Line of Sight (BLOS), the requirements additionally need to be fulfilled differently. For example, with direct communication for LOS or a more complex communication utilizing satellites for BLOS. [82].

In contrast to UAVs, tethered systems for underwater missions, called ROUVs, exist. An example of a ROUV named *Hercules* can be seen in Figure 1.3. These systems "play an

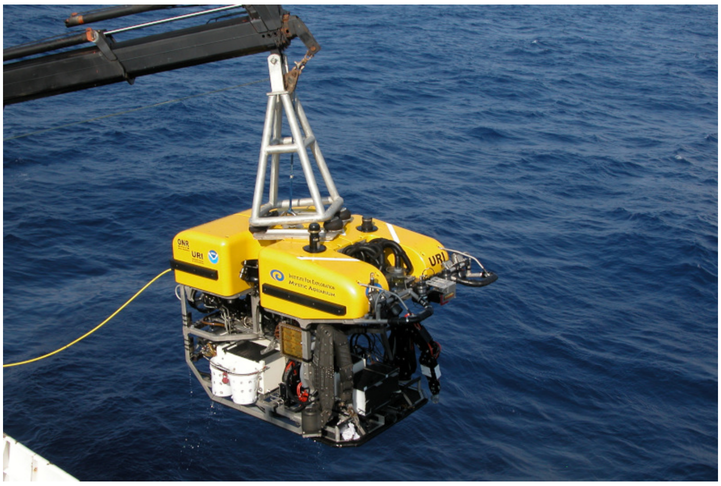


Figure 1.3: The research Remotely Operated Underwater Vehicle (ROUV) Hercules (Source: [22]).

important role in a number of shallow and deep-water missions for marine science, oil and gas extraction, exploration and salvage" [61]. They are usually controlled by a human operator on a ship above water surface. Control can either be direct or on a higher level to provide the ROUV with greater areas of freedom, which means that the operator can either manually control the ROUV or provide waypoints that the ROUV navigates to autonomously.

For ROUVs, the potential drawbacks of remote vehicles like high latency, unstable connection and low bandwidth are usually not present as the vessels are typically connected by an umbilical cable. [61] For heavy systems they can reach a depth of more than 3000 m, however, depending on the capabilities most ROUVs operate in a depth of up to 300 m. Thus, a common classification of ROUVs is based on their capabilities, as shown by Azis et al. in [17].

The development of Unmanned Ground Vehicles (UGVs) has a long history [60]. Such vehicles operate on ground and can be controlled either wireless or tethered [97, 60]. Like previously introduced systems, the UGVs provide different levels of abstraction and autonomy for its maneuvering. The operator then only provides waypoints or higher level tasks [136]. Their usage scenarios are widespread and range from military operations [60] to search and rescue [97] and inspection of specific construction elements [73]. They are mostly used in dangerous or hazardous situations [136]. Figure 1.4 shows an example of such an UGV, the *Leopardo B*, which for example can support in investigation and rendering harmless dangerous objects [53].

Figure 1.4: The Leopardo B UGV of EuroLink Systems (Source: [141]).

1.1.2 Components of Teleoperated Systems

All above introduced systems consist of few basic shared components. Following Winfield [176], the three basic components of every teleoperated system are:

- the **robot** (remote vehicle),
- the **remote place of work** (teleoperation station),
- the **connection** between both

The **robot**, the remote vehicle, consists of the parts that are required to fulfill its designated task, e. g., driving on streets, flying, sensing. It is usually extended by specific hardware and software to enable remote controlling. The main part of such a system is the central computing component. This computing component is responsible for handling various tasks such as dealing with sensor inputs and control commands [**IV**]. A communication interface for exchanging data with the remote driver is required to enable the connection. Depending on how the system to be controlled remotely is already equipped, certain modifications are required. Those modifications consist of remote control-specific sensors combined with task-related ones, e. g., additional video cameras, ultrasonic sensors might be required. [176]

The **remote place of work** supplies the human operator, also called teleoperator, with information about the remote vehicle and the corresponding environment and enables him to control the vehicle remotely. Thus, the remote place of work has to provide output interfaces such as monitors, head-mounted displays, actuators for haptic feedback, etc. as well as input interfaces such as controllers. The decision of which interfaces and controllers are required highly depends on the remote robot and the task to be executed. [176]

The **connection** between the robot and the remote place of work is used for exchanging data. It could be either tethered or wireless depending on the use cases. Transmitted data consists of sensor information that either track the environment or the robot's state and control data from the remote place of work. The type of connection as well as the content of the transmitted data mainly depends on the setup and task. [176]

Beside the requirements of the task to be fulfilled, a major influence on the architecture comes from the level of autonomy the robot should provide. Based on the level of autonomy, excluding fully autonomous systems, the control can be split into two major categories, direct and supervisory control [157]. By using direct control systems, the operator directly controls the system. In contrast, by using supervisory control, the operator sends commands to the remote device which executes them itself while being supervised by the operator. [157]

1.1.3 On-Road Teleoperated Driving

The basic concept described in [176] can be mapped to typical street vehicles. An example setup of such a system with remote vehicle, teleoperation station and the connection between both can be seen in Figure 1.5.

On the left side of Figure 1.5, the remote vehicle with its components is shown. Basically it contains a car PC, a 4G/5G modem and a bus system connecting the sensors and actuators. Typical sensors are cameras, Light Detection and Ranging (LiDAR), radars and ultrasonic sensors, which may be required in different driving situations.

On the right side of Figure 1.5, the teleoperation station with its displays and controls can be seen. It mainly consists of a steering wheel and pedals as well as some output interfaces like

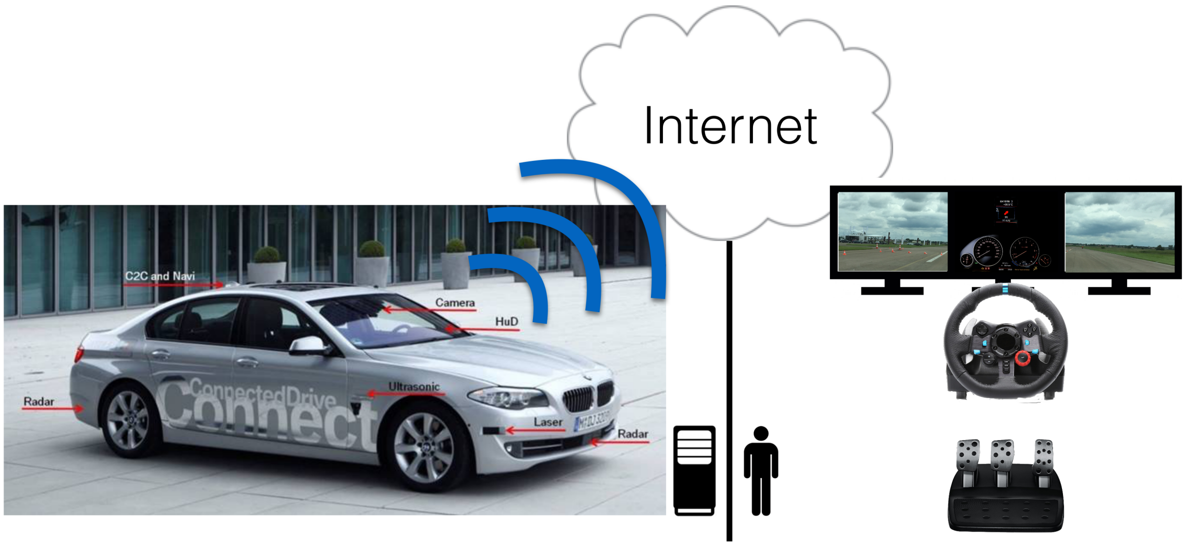


Figure 1.5: Teleoperated system, with the vehicle on the left side and the teleoperation station on the right side. (Based on [165], Source: [**II**], Reproduced with permission from Springer Nature)

monitors or virtual reality equipment [68]. The station can also be equipped with haptic feedback [87]. The vehicle and the teleoperation station are connected remotely. For teleoperated driving the use of cellular networks is one of the most suitable approaches [**IV**].

Following the definition of [177] and extending the direct and supervisory control approach, teleoperated driving can be categorized into four groups, based on the role and responsibilities of the teleoperator as can be seen in Figure 1.6. Responsibilities are split into strategic operation (planning), tactical operation (detect and respond) and operational operation (control vehicle). For **ToD type 0**, the teleoperator is not engaged in any aspect of the actual driving, while in **ToD type 1** the teleoperator acts as a dispatcher. Starting from **ToD type 2**, the teleoperator is responsible of performing the indirect control of the vehicle (e. g., specify obstacle avoidance path), while finally in **ToD type 3** he directly controls the vehicle. [177] The focus of this work is on type 3, the direct control, where the teleoperator's steering commands are directly transferred to the vehicle and executed accordingly.

When operating a remote vehicle, there is a permanent exchange of information and control commands. Sensor data and vehicle information is continuously transmitted to the teleoperation station, where it is displayed. The teleoperator permanently monitors the vehicle's state

ToD Type (Role of ToD operator)	**Act of Driving**		
	Strategic Operation (Travel planning, route and itinerary selection)	Dynamic Driving Task (DDT)	
		Tactical Operation (Object and Event Detection and Response OEDR)	Operational Operation (Sustained lateral and longitudinal vehicle motion control)
0 (No Role)	In-vehicle user or system	In-vehicle user or system	In-vehicle user or system
1 (Dispatcher)	ToD operator	In-vehicle user or system	In-vehicle user or system
2 (Indirect Controller)	ToD operator	ToD operator	In-vehicle user or system
3 (Direct Controller)	ToD operator	ToD operator	ToD operator

Figure 1.6: Operator's tasks based on the different levels split by strategic operations, tatictal operation and operational operation (Source: [177]).

together with the vehicle's environment and reacts accordingly by providing steering commands. These steering commands are transferred to the remote vehicle, which then executes them.

In contrast to real-world sized teleoperated vehicles, e. g., as in Figure 1.5, there are also more compact ones that can act as demonstrators or prototypes. One example is introduced by [24]. In such small-scale demonstrators, communication is possible with a broader range of different technologies, usually offering the possibility to introduce real-world effects, e. g., latency, bandwidth limitations, packet loss, to evaluate real-world teleoperated driving situations in a miniaturistic way. This allows to test algorithms or technologies in a controlled environment. One example is conducting a user study without the risks of destroying an expensive car or causing injuries.

Opportunities of Teleoperated Driving

The use cases of teleoperated driving are widespread and mainly depend on the level of automation the remotely controlled vehicle can offer, e. g., as shown in [**IV**]. An example for the use of teleoperated driving is valet parking. A human driver drives the vehicle to the desired destination. If the vehicle has to be parked, a remote driver takes over control and executes the task. Thus, it is not necessary to have a driver inside the vehicle during the parking process. The remote driver will also drive the vehicle back to pick up the "passenger" again. The "passenger" then takes over control again and steers the vehicle himself. Another example is the use of teleoperated driving for car rentals or car sharing services. In case of car sharing services, it is possible to place vehicles remotely, without requiring an employee to drive the vehicles physically. In car rentals, the overall fleet management could be eased, e. g., in case of One-Way-Rentals. In [32] it has been shown that the number of taxi drivers could be reduced by 15 to 39 percent if the vehicles are controlled remotely through teleoperated driving. Even with full autonomous vehicles, teleoperated driving can be vital. With an increasing number of sensors and more complex algorithms, errors or undefined situations are more likely to occur. As an example, complex road side works could stop an autonomous vehicle. A remote operator might take over and resolve the issue. In general, teleoperated driving can always be used if the vehicle is not able to solve a situation itself, as humans nowadays are better in handling unknown traffic situations compared to machines [14]. Teleoperated driving is already deployed for the delivery robots of *Starship* [101] or used within the autonomous vehicles of *Waymo* [39]. Teleoperated driving could also be used to reduce the fear in autonomous driving features. A survey of Bitkom Research has shown that people are skeptical regarding autonomous driving features [21]. If the passengers knew, that they are driven by humans, this skepticism could potentially be lowered, as this additionally offers the opportunity of communicating with the remote operator. Further use cases of teleoperated driving are conceivable. Especially in the area of logistics as shown by *Einride* [77] with autonomous trucks and the according possibility for remote operation or *Fernride* [57] with their remote operation platform for logistics. Thus, teleoperated driving is claimed to play an important role in the development, spread and acceptance of autonomous vehicles. Although, only foreseen as a system for supervision without any abilities for providing steering command beyond the activation of an emergency mode, the concept of teleoperation is already mentioned by the German government [23].

Problem Description

Teleoperated driving offers the opportunity to allow skilled human drivers to remotely control a vehicle whenever it is required. Unfortunately, it is not that trivial to directly control cars from a remote teleoperation station. In teleoperated driving, maneuvering has to be done based purely on the information provided by the vehicle's sensors. In comparison to physically driving a vehicle, however, the teleoperator is only provided a subset of the total information about the traffic scenario and environment due to the sensors' limitations. The teleoperator usually only has access to video streams and additional supporting sensor data. This makes it hard to safely control the car in different environmental situations. Even if a teleoperator is able to handle the remote vehicle properly, there are further issues regarding the link between teleoperator and remote vehicle. Examples are high latency, limited bandwidth and packet loss. A technology that is already widely deployed has to be used to allow remote operation over a broad geographic area. Cellular networks could become the technology of choice, since they are able to meet required latency, bandwidth and packet loss demands [**VII**]. Further technologies such as satellite based communication or Wi-Fi covering large areas are currently too expensive and as such not feasible. Unfortunately, the indispensable use of wireless technology will always go hand in hand with reliability-related issues, emerging from the required stable communication between teleoperator and remote vehicle. Depending on various parameters, the performance of the connection can vary drastically, making teleoperated driving challenging. With respect to safety and comfort, latency, bandwidth and packet loss are crucial.

In teleoperated driving, all three components, **robot**, **remote working place** and **connection** are essential, however, the focus of this work is on the **connection** part. In order to ensure a safe remote control, the connection has to meet specific requirements.

A typical setup for teleoperated driving consists of a remote vehicle equipped with a LTE/5G modem [33] and a teleoperation station which is connected to the internet. Preferably, the teleoperation station is connected to the internet by wire due to lower the effects of wireless connections. The vehicle and the teleoperation station are thus connected via Internet connection. It is hardly possible to apply fundamental changes to this whole infrastructure and thus it has to be treated as a black box. This induces communication challenges, e. g., as shown in [28, 32] that are explained in more detail in the following. For being able to safely control a vehicle remotely in real-world traffic scenarios, it is important that the teleoperator is able to always know the state of the remote vehicle and its environment to be able to react to expected and unexpected obstacles appropriately. Due to the nondeterminism of the connection through different types of networks, this is hard to ensure. In general, teleoperated driving is facing four major Quality of Service (QoS) challenges: bandwidth, latency and jitter as well as packet loss. Because of the high mobility of vehicles, a higher frequency in the changes of network conditions is expected. This leads to a situation, in which the QoS is even harder to ensure.

Uplink: Sufficient uplink is required to transmit sensor information from the remote vehicle to the teleoperation station. Streaming sensor-data like video or LiDAR content requires a huge amount of uplink on the remote vehicle's side. If uplink is not sufficient, the teleoperator does not get all required information of the remote vehicle and its surroundings and thus might not see emerging obstacles or other important details. [**VII**]

Downlink: Downlink is required to receive the control commands that are provided by the teleoperator. The required amount of data that needs to be transferred in this direction can be kept considerably low. If downlink is not sufficient, the remote vehicle is not able to receive steering commands and thus cannot be controlled safely. [**VII**] In contrast to many other applications such as watching video streams on prominent platforms, the requirements regarding uplink and downlink are inverted for teleoperated driving.

Latency: Latency is always present and introduced not only by the network but also by sensors, actuators and processing steps. It may result in required information arriving too late. Information in this context means either control commands or environmental/sensor data of the remote vehicle. Vehicles thus might end up in dangerous situations, as the operator sees the changes in the environment too late, steering commands at the vehicle are received too late or a combination of both. In the worst case, this can result in a severe accident.

Variable latency, Jitter: Jitter makes it even harder to safely control a remote vehicle. If constant latency is present, a teleoperator might be able to deal with it by adjusting its driving. But if latency is varying, it is hard to foresee the impact and adjust the driving. [**VII**]

Packet Loss: When using network technologies, especially wireless ones, a high probability of packet loss exists. In the worst case this can lead to a temporary complete loss of information. This issue can be addressed by using protocols that resend packets. However, resending packets adds additional latency to the connection and thus a trade-off between reliability and latency needs to be done. In general both cases can negatively impact the ability to safely control the remote vehicle. A total loss of packets would leave the teleoperator “blind”. However, a delayed delivery of steering commands (through resending) could lead to a situation in which the remote vehicle acts inappropriate to its environment. As can be seen, both examples are extremely bad and a solution to this trade-off needs to be found. Two well known examples that face either packet loss or additional latency are User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). Using unreliable protocols such as UDP, packet loss can lead to information loss. In case of reliable protocols such as TCP, latency may be increased by the process of waiting on information that needs to be retransmitted.

In order to make the system work in spite of the aforementioned challenging areas, this work deals with the respective impacts and presents potential solutions. Those solutions can help to reduce or completely overcome these obstacles, allowing to safely enable teleoperated driving in everyday’s traffic scenarios.

1.2 Contribution of this Thesis

Based on the main research claim of this thesis – *enabling teleoperated driving in everyday’s traffic scenarios* – our major contribution is to provide solutions that help to alleviate the impact of the network issues with contemporary cellular networks on teleoperated driving.

Therefore, we address the challenges with real-world measurements, system design proposals and different technological approaches that make the system more usable. More specifically, we present the major influence of bandwidth and latency on teleoperated driving combined with potential solutions to overcome these obstacles. Thus we address the following research questions:

RQ1 Is teleoperated driving feasible with contemporary cellular networks?

RQ2 Is it possible to achieve a safe remote control of a vehicle on typical roads despite latency using basic algorithms?

RQ3 What is the influence of variable and fixed latency on teleoperated driving performance and subjective assessment?

RQ4 How far can video streams be compressed to still allow for safe teleoperated driving?

RQ5 How can the required bandwidth for uplink video streams be decreased?

In order to answer the question whether teleoperated driving is feasible at all with contemporary cellular networks (**RQ1**), in Section 3.1.2 as published in [**VII**] we present real-world cellular measurements. Those measurements are focused on latency and bandwidth. Thus, we applied different measurement approaches during real-world drives, i. e., measuring Internet Control Message Protocol (ICMP), UDP latency, UDP bandwidth and TCP bandwidth. The contents of the work presented in Section 3.2.5 as published in [**III**] address the research question **RQ2**. We introduce an innovative way on how to safely handle different latencies in teleoperated driving. As such we provide the design of a driver support system and a route planning algorithm that are based on the main idea that high latency can be counteracted by reduced remote vehicle's speed. With the growth of testing complex algorithms for teleoperated driving, the need of virtual testing arises. Lacking a suitable driving simulator on market, we developed OpenROUTS3D for the needs of teloperated driving. Based on the work published in [**V**], Section 3.3.3 introduces OpenROUTS3D, explains its fundamentals and illustrates its features. Utilizing the previously introduced driving simulator, Section 3.4.4 (originally published in [**VIII**]) addresses **RQ3** by presenting a user study, in which participants had to drive different scenarios of teleoperated driving with different levels of latency. Based on subjective and objective measurements we identified suitable levels of latency. The following Section 3.5.3 presents the results of the work published in [**VI**], where we address **RQ4** by examining different video stream qualities. We evaluated the quality of real-world driving clips with a user study, allowing to identify the minimum quality requirements for each stream and map this to respective bandwidth requirements. An approach to further lower the bandwidth requirements (**RQ5**) is lastly proposed in Section 3.6.4, which is based on the work [**I**]. The main idea is that only important objects in the stream need to stay sharp and fully detailed, while the remainder can be manipulated in a way that less bandwidth is required. We present the results of a respective user study and finally propose a supporting system addressing the previous results.

1.3 Structure of this Thesis

Following the introduction of teleoperated driving in **Chaper 1**, **Chapter 2** presents related work with respect to the contents of the presented publications. Subsequently, **Chapter 3** shows the contributions of this thesis. Finally, **Chapter 4** summarizes the results of the presented work, discusses its the limitations and presents potential topics for future work.

2 Related Work

The majority of research in the area of teleoperated driving focuses on the impact of datarate and latency, visualization of data at the operator's working place and general usage studies for teleoperated vehicles. A survey done by Lichiardopol [113] provides a general overview of teleoperation, while addressing important aspects such as telepresence and control issues. Chen et al. [26] executed another detailed investigation of about 150 papers, where their focus was on further motivating research by pointing out specific areas that may benefit from it. They address a broad area of different topics for human performance issues and provide potential mitigations for remote operations.

The portion of related work within this thesis focuses on areas of teleoperated driving, which are closely related to the research we present in Chapter 3. Therefore, our addressed topics are *Network*, *Teleoperated Driving Systems* and *HMI with User Studies*. Although the related work has been subdivided into different topics, a strict distinction is not possible. Nevertheless, addressing those partly overlapping topics helps to get an overview of what was already done in the area of teleoperated driving and where research gaps exist.

2.1 Network

One of the biggest obstacles of teleoperated driving is the network link between the remote vehicle and the operator. In order to make teleoperated driving feasible, it is important to use already widely deployed technology such as cellular networks with the respective technologies, e. g., Long Term Evolution (LTE) and 5G.

Previous work put a lot of effort into this area, investigating the impacts of bandwidth and latency on teleoperated driving. Latency is measured differently (one-way or Round Trip Time (RTT)) on the presented papers and thus the type of the conducted measurement is stated.

Therefore, different measurements were conducted. Chucholowski et al. [28] measured the one-way latency of video stream in 3G networks while driving. With a highly varying average RTT of 121 ms, ranging from 65 ms to 1299 ms, they claim that 3G connections can already be sufficient for remote driving scenarios. Kang et al. [93] revealed an LTE-based two-way latency of 100 ms if transmitting a video-stream, using a real-world testbed for analyzing LTE and IEEE 802.11 Wireless Local Area Network (WLAN). Further measurements were conducted by Shen et al. [156], who saw an average latency for video streams (one-way) being about 183 ms for 4G and 205 ms for 3G, with a range of 119 ms to 463 ms. Additionally, they investigated the vehicle control latency (one-way), with an average latency of 110 ms for 4G and 217 ms for 3G. 21 ms was the minimal latency while 677 ms was the maximum. It turned out that, although 3G could provide lower minimal latency values than 4G, it is less stable and has higher maximum values.

Keon Jang et al. [92] conducted measurements in 3G and 3.5G networks during driving a car (100 km/h), riding high speed trains (300 km/h) and standing still, i. e., doing stationary measurements. A significant difference between mobile (car and trains) and stationary measurements could be seen. Lower throughput over UDP and TCP as well as higher jitter and packet loss occurred in the mobility scenarios. The cellular network measurements of Xiao et al. [180] were conducted at speeds above 300 km/h and they compared them to the results of stationary and mobility measurements below 100 km/h. As such they drove a car for 120 km and rode a high speed train for 5000 km measuring throughput and latency. It turned out that TCP throughput and RTT are worse in the high speed mobility scenario, compared to the stationary and mobility measurements. However, one of their conclusions is that the variance is more affected than the absolute values. Lauridsen et al. [107] drove about 19,000 km in a mix of rural, suburban and urban environments. During the driving they measured latency, handover execution time and coverage of four operational LTE networks. Their results revealed a LTE coverage of about 99% combined with an average handover latency of about 40 ms. Measurements of Merz et al. [124] indicate that LTE performs robustly up to 200 km/h, if the coverage of signal-to-noise ratio (SNR) is “well dimensioned” [124]. Inam et al. [89] conducted LTE measurements by streaming a video in one cellular network cell to check whether the latency is suitable for teleoperated driving. They identified that it is suitable in most of the measured areas in Sweden. In about 98% of their measurements, the latency was below

50 ms.

Yu and Lee [181] measured the real-world latency in South Korea with LTE and WLAN setups. They showed that due to lower RTT values, WLAN would serve remote driving better, while their finding claims that latency is not necessarily proportional to the geographical distances between source and target. Toril et al. [168] present an on-road analysis of the radio signal strength fluctuations with focus on LTE. They measured 1000 km of driven road with two vehicles sharing an identical measurement setup on the same route at different times, in order to compare fluctuations on the radio signal level. They claim that even different driving lanes may receive different signal levels. Induced by handovers between two network cells, they measured a maximum difference of 23 dB at the same position, that could have influence on specific ultra-reliable low-latency services, e. g., teleoperated driving.

The work of Li et al. [111] consists of a comparison between CUBIC, Bottleneck Bandwidth and Round-trip propagation time (BBR) TCP congestion control driving on a highway. They measured RTT latency using ICMP, TCP connect times and throughput by downloading a file from a server. They utilized a smartphone to conduct those measurements and witnessed a predominant latency between 40 ms and 80 ms, while the TCP throughput was at about 11 Mbps in median. LTE uplink results of Parichehreh et al. [142] show that BBR works as expected, but on device packet losses can be seen. In [24], Burke developed a micro-rover that could be controlled either through WLAN or based on the 4G network. He described the basic system setup and demonstrated that it was possible to manually control the rover remotely on sidewalks and during a 1.5 h drive, the connection only dropped three times. However, the work also identified a stream-to-monitor latency of more than 700 ms in 4G, while WLAN only had 200 ms. The work of Georg et al. [64] analyzes the end-to-end latency for teleoperated driving by splitting the influence into actuator and sensor induced latency. They showed that the latency between steering input at the operator's side and actual change of the steering wheel's position at the remote vehicle takes about 80 ms using LTE for singal transmission. 60 ms out of this are required to turn Controller Area Network (CAN) signals into an actual change of the steering wheel's position, i. e., the major latency is induced by the actuator. With a sensor latency of about 121 ms being introduced mostly by the network and the used monitor and not by the camera or processing, a total end-to-end system latency of 190 ms could be achieved within their setup.

Mouawad et al. [126] present a mobility management scheme for teleoperated driving by focusing on Software Defined Networking (SDN). Their approach is based on handover anticipation and route proposal in order to allow for a seamless handover via two interfaces at the remote vehicle. Their results show improvements in reduced network load, lower handover, signaling delays and zero packet loss ratio. In general, it can be said that the mobile connections highly suffer from potentially high delays, variable bandwidth and packet loss. [42]. Already with LTE-Advanced the uplink rate is increased to up to 1.5 Gbps [174], which should be enough to transmit required data and commands in teleoperated driving use cases. With 5G, which is currently becoming more and more widespread, even better values can be achieved [69].

Although 5G is claimed to mitigate existing problems, e. g., by using dedicated communication channels [150], the first 5G installations initially only cover a limited area. Thus, their widespread application will take its time. In addition, measurements will be required in order to investigate whether the promised capabilities can be achieved to support teleoperated driving.

Despite this huge amount of measurements and respective research, a lack of large-scale measurements with specific focus on teleoperated driving could be identified. It is important to identify whether teleoperated driving is feasible with contemporary networks and which influencing factors and network bounds need to be considered in real-world applications. Thus, we address this gap with our contributions to **RQ1**, where we conducted real-world driving measurements in Germany to measure the cellular network performance.

2.2 Teleoperated Driving Systems

Research in teleoperated driving also puts some focus on the development of solutions to overcome the system's barriers, e. g., latency and bandwidth-related issues.

The majority of approaches target the improvement of the situational awareness for the remote operator, which will be achieved best if he has access to all relevant environmental information [135]. One approach is the introduction of predictive displays, as done by Davis et al. [40], that are used to show the path and position of the vehicle according to the latency, which helps to allow for a safer driving. Chucholowski [29] compared various predictive displays and reached the conclusion that such systems can efficiently support the operator. The work of Graf et al. [70] improves the predictive displays by including both, the remote vehicle inputs, the latency and – this is the improvement – the operator inputs. It is also shown in [85], that the use of virtual reality glasses in combination with available sensor data can help to improve the situational awareness, i. e., by merging camera-based video streams with 3D models created based on LiDAR data. Another way to improve the situational awareness is the utilization of haptic feedback as if driving a car in the real world, e. g., as proposed by Hosseini et al. [87].

Further developments aim at improving situations that have bad network connectivity. As shown by Hosseini and Lienkamp [86], one feasible approach is to equip the remotely operated vehicle with the ability to autonomously react to yet unknown upcoming hazards, e. g., by adjusting the speed. A different method is presented by Tang et al., where a free corridor is set by the remote operator in order to provide a path for the next few seconds, i. e., if network connectivity drops, the vehicle is still able to move safely on that path [166]. Schitz et al. [153] present a corridor-based approach where they combine corridors defined by operators and automated driving functions, in which the operator provides the specific corridor, while an algorithm will determine a collision-free path based on sensor information. Schimpe and Diermeyer present a new control approach [151], in which the steering reference of the human remote operator is tracked and rectified with potential fields if driving actions have the change of a collision.

Additional proposed approaches put their focus especially on mitigating latency and bandwidth issues on the system itself. Luck et al. [119] investigated the effects of different types of latency on the performance of remote operation. Their claim is that it is easier to deal with constant latency than with variable one. This finding is confirmed by the results of Davies et al. [40] and Liu et al. [115]. One way to achieve constant latency is introducing buffers as already done for years in video streams [158] and also proposed for teleoperated driving by e. g., d'Orey et al. [32]. It is also possible to use multiple SIM-cards of different providers [75] to always select the best overall possible network at the current location, as different providers have different areas of coverage with their cellular network. One way to keep latency and bandwidth consumption as low as possible is the use of unreliable protocols like UDP, which can help to reduce the communication overhead [28]. In order to address the issues of bandwidth consumption a lot of research has already been carried out on dynamic rate adaptation [103]. One approach with focus on teleoperated driving was presented by Gnatzig et al. [68]. Based on heuristics, they showed that compression parameters can be adjusted to the available bandwidth. For the front-camera, two different setups utilizing H.264 were compared. The first one had a resolution of 640x480 and a Constant Rate Factor (CRF) of 25, which led to a bandwidth consumption of about 1678 kbps. The second setup used a resolution of 320x240 with a CRF of 30, which led to 222 kbps. [68] Hofbauer et al. [83] present an approach which helps to address the fact, that individual cameras on a remote vehicle may requires different levels of quality in the stream. With limited hardware capabilities, e. g., single hardware-encoder, on the vehicle this was a complex task. Using preprocessing filters the authors overcame this problem by allowing for quality adaptions in the superframe video, which is the video consisting of all camera's contents. The work presented in [71] reduced the overall bandwidth-requirements to about 15 kbps by only transmitting a compressed version of captured point cloud data.

Although different approaches to overcome the issues that are induced by latency and bandwidth exist, multiple gaps could be identified. One gap is based on the relationship between remote vehicle's speed and the respective latency. We address this gap with the work answering **RQ2**, where we apply speed reduction to counter the negative influence of latency. Another gap that could be identified considers available uplink. Although different compression and encoding techniques were applied to reduce the bandwidth consumption, a more thorough approach on video streams could be considered (**RQ4** and **RQ5**). In order to answer **RQ4**, we conducted a user study in which participants rated different video-stream qualities. This helps to identify the minimum required quality and therefore the respective minimum bandwidth for a specific driving situation. By answering **RQ5**, we show that bandwidth requirements can be reduced by keeping important areas of a video stream sharp, while less important areas can be manipulated in a bandwidth friendly way.

2.3 HMI with User Studies

Another aspect for teleoperated driving is the ability to provide the so called telepresence, which means to provide a feeling of the remote vehicle and its environment [25]. In general, it can be said that ideal situational awareness can be achieved if the operator is able to see all information of the relevant environment [135]. Ideal approaches and potential system bounds need to be investigated by conducting user studies.

Hosseini and Lienkamp [85] show that the performance of teleoperated driving could be improved by virtual-reality systems, if combining the available sensor data. However, in contrast to these results, the comparison between a multi-monitor setup and Head-Mounted Display (HMD) done by Georg et al. [63], showed that such a HMD allows for a better immersion, but does not necessarily increase the driving performance. In their work [44] Doki et al. utilize a HMD to allow for the 360 degree presentation of the content of a remotely controlled robot, which is equipped with a 360 degree camera. Based on LiDAR data, they augmented the stream in order to additionally project 3D objects into the HMD. Their results show that the object projection can help to counteract for delays in the stream or dropped packets, i. e., the task performance was improved. In [65] Georg et al. conducted a five week long user study with 30 participants in order to investigate the influence of different display methods, stream qualities and video canvases on objective and subjective measures such as perceived workload, usability, etc. Their objective measures indicate that the quality of the video stream has an influence on the situational awareness and participants liked head-mounted displays better compared to typical setups. However, results on conventional monitors were comparable and participants were able to identify important objects on all tested display types. Schimpe et al. [152] address the topic of adjusting video streams dynamically to a given uplink bandwidth. Based on rate-quality models for individual cameras, their approach adjusts the video quality with respect to the resolutions and the dynamically allocated bitrates. Langer and Topp did a comparison of three different displays for remote supervision in their work [105], spending attention especially on safe navigation and situational awareness by keeping the cognitive load low. By conducting a user study, they showed that enhanced 3D representation and Virtual Reality (VR) representations are efficient ways to keep track of the environment and are rated better in comparison to traditional representations. Nielsen et al. [135] conducted a user study where a combined 3D view consisting of video, map and robot-pose information was investigated. Although this work was not specifically done for teleoperated driving, the results show that the introduced approach can improve the driving performance. Utilization of additional haptic feedback on the steering wheel was investigated by Hosseini et al. [87]. Their results show, that haptic feedback can help to improve the telepresence and thus the driving performance. Hosseini and Lienkamp [86] present a safety concept for teleoperated driving in which the remote vehicle automatically reacts on known and due to time delay yet unknown hazards by speed reduction. Their results show that the proposed approach can help to increase driving safety.

Liu et al. [115] utilized a small-scale vehicle to conduct a user study with state-of-the-art network performance of LTE. Their claim is that teleoperated driving on such networks does not work if supporting systems that help to safely control a vehicle remotely do not exist. However, Shen et al. [156] show, that driving a slalom course with a usual car that was turned into a remotely controlled one with off-the-shelf components was even possible with 3G network connectivity. The user study of Vozar and Tilbury [173] was conducted to explore the effects of latency in teleoperated driving, showing that the path-following scores become worse if latency is increased. Gnatzig et al. [68] show some results of a real-world test drive. They find that, with a latency of about 500 ms and 30 km/h the vehicle could be safely controlled and react to dynamic objects in a feasible manner. Jahromi et al. [91] conducted a user study in order to investigate the ability of participants to perform typical remote tasks related to navigation by addressing it via Quality of Experience (QoE). They altered typical influencing network parameters such as latency, bandwidth and packet loss. Results indicate that participants were able to differentiate between visual aspects and navigation or control and both aspects have a task-depend weak correlation. Dybvik et al. [48] performed a user study with 57 participants to investigate, whether the utilization of a predictive display can help to increase driving performance and reduce the perceived workload under different latency with 250 ms for baseline and 700 ms with and without predictive display for measurement. Their results indicate a 20% increase of objective driving performance ("course completion time and task score" [48]) utilizing the predictive display.

In addition to real-world tests, using driving simulators is crucial, as this allows for reproducibility and the test of new algorithms and technologies in a safe environment. Therefore, some of the best known and potentially suitable driving simulators are presented in the following. One proprietary piece of software for conducting virtual test drives is the tool CarMaker [15], which can be used in multiple steps of the development process by offering the ability to transfer real-world scenarios to highly detailed virtual test-drives. With Automotive Data and Time-Triggered Framework (ADTF) [50] a frame work exists, that supports in the development process of autonomous software. The strength of AVL Cruise [16] is the simulation of a vehicle's driveline by using it within Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) testbeds. In the class of Open Source driving simulators various tools exists. One well known tool is Open Source Driving Simulation (OpenDS) [121], which is platform independent, GNU General Public License (GPL)-licensed and Java-based. Being platform-independent, it provides analysis and simulator functionality such as traffic simulation, different environmental conditions and highly detailed cars. With CARLA, a MIT-licensed autonomous driving simulator exists [45]. Based on a flexible API, Robot Operating System (ROS) integration, sensor simulation and the dynamic map generated based on the OpenDrive standard [47], it allows for different types of research and development in the field of autonomous driving. Based on the Unreal engine and developed by Microsoft, AirSim [155] is a platform that allows for experimental use of AI such as computer vision. The Self-Driving Car Simulator of Udacity [170] was developed to allow the training of autonomous vehicles. Multi-Agent DRiving Simulator (MADRaS) [149] is an improvement of the TORCS racing simulator [179], which can be used as research platform. Finally, with Deepdrive [146] an end-to-end simulator for autonomous vehicles exists. With respect to teleoperated driving, SILAB was used in [85] and [87], while DYNA4 was used by [29]. SILAB [178] has its focus on the interaction between driver, vehicle and traffic while DYNA4 [172] is designed for function development in closed-loop systems ranging from powertrains to traffic.

Considering existing related work, two gaps could be identified. One is the lack of a driving simulator that is focused on the needs for teleoperated driving research. Thus, we developed Open Realtime OSM- and Unity-based Traffic Simulator 3D (OpenROUTS3D), a driving simulator that is tailored to the needs of teleoperated driving. Although a lot of user studies were conducted, none of them addressed objective and subjective metrics when driving a vehicle remotely on different everyday driving situations. In order to answer **RQ3**, we conducted a user study. Participants had to drive different scenarios virtually on OpenROUTS3D and answer specific questions to measure their subjective workload accordingly.

2.4 Summary

With this extract of related work, it can be seen that a lot of effort is put into research with focus on teleoperated driving. The latency and bandwidth capabilities of different cellular networks are measured and analyzed with various methods. Depending on the results, findings show that teleoperated driving could be already possible with 3G networks. It was also shown that not only the network-induced latency is crucial, but also the one that is added by sensors and actuators. In order to overcome those obstacles, research focused on multiple solutions. Those solutions map well known concepts such as buffering to reduce jitter and video rate adaptation techniques to reduce bandwidth consumption to teleoperated driving, e. g., such as predictive displays that help to show the actual position of a remote vehicle considering latency. Furthermore, research also focused on achieving telepresence, i. e., enabling the teleoperator to get a feeling of the remote vehicle and its environment. This is done by testing different types of displays and feedback systems. In order to test applications of teleoperated driving, different driving simulators were already used for investigating various aspects of teleoperated driving. As shown within the respective sections, different gaps could be identified and we addressed them by answering **RQ1** - **RQ5**. Large scale, real-world driving measurements with focus on teleoperated driving were missing and thus addressed by **RQ1**. Latency-based speed adjustments were identified as being helpful for teleoperated driving, but were not properly covered within related work. Thus, we investigated latency-based speed adjustments in order to answer **RQ2**. Video streams were considered a lot, but a deep understanding of minimal qualities and respective bandwidth requirements was missing. In order to answer **RQ4**, we carried out a user study where participants had to rate the perceived video-quality. This helped to obtain minimum bandwidth requirements. Taking this approach a step further, we answered **RQ5** by applying an approach that is based on object detection. Important areas of a stream are kept sharp, while less important ones are manipulated in a way that helps the video stream encoder to reduce bandwidth requirements. It also turned out the no suitable driving simulator for teleoperated driving existed. Thus, we developed OpenROUTS3D. Finally, no user study that considers the impact of latency on typical driving tasks existed. Therefore, we carried out a virtual user study (OpenROUTS3D) to help answering **RQ5** by measuring objective and subjective metrics driving a car remotely in different scenarios.

3 Teleoperated Driving System Design and Evaluation

Although a lot of research has already been done with respect to teleoperated driving, there are still important questions which have not been addressed and answered. Considering network-induced issues as the main focus of this thesis, the before mentioned research gap is addressed by answering the following research questions **RQ1–RQ5**:

RQ1 Is teleoperated driving feasible with contemporary cellular networks?

RQ2 Is it possible to achieve a safe remote control of a vehicle on typical roads despite latency using basic algorithms?

RQ3 What is the influence of variable and fixed latency on teleoperated driving performance and subjective assessment?

RQ4 How far can video streams be compressed to still allow for safe teleoperated driving?

RQ5 How can the required bandwidth for uplink video streams be decreased?

With teleoperated driving being a system that is used in large geographical areas, the use of cellular networks cannot be bypassed, as other solutions might not be feasible or too expensive, e. g., an own network dedicated for teleoperated driving. Hence, one important step is measuring contemporary cellular networks during real-world driving to identify whether these networks are able to deal with the network requirements for teleoperated driving. Therefore, we conducted real-world measurements [**VII**] to provide a first assessment on whether contemporary networks are feasible for teleoperated driving. We measured latency and throughput values using three different setups. Additionally, different parameters such as handover, distance between a remote car and the teleoperation station and signal strength were analyzed to see whether an influence on latency and throughput exists.

In order to deal with latency that is induced by the network, sensors and actuators, an algorithmic approach to lower the impact is crucial. Therefore, we present a supporting system for dealing with latency-related issues (published in [**III**]). The main idea is that the negative influence of latency could be compensated by reducing the remote vehicle's speed in order to maintain a safe state for the controlling teleoperator and the vehicle.

To be able to conduct user studies for teleoperated driving, the use of simulations is indispensable. It helps to reduce costs, lowers the risk of damaging expensive parts or harming someone. Additionally, it helps to tackle the issues of reproducibility, which is an important part when conducting user studies. Because existing simulations did not offer sufficient functionality for teleoperated driving, were too complex to enhance or too expensive, we decided to develop a driving simulator, called Open Realtime OSM- and Unity-based Traffic Simulator 3D (OpenROUTS3D)[1]. This driving simulator is introduced in [**V**]. It is publicly available as Open Source under GNU General Public License (GPL) v3 and designed to fit the needs of teleoperated driving. OpenROUTS3D provides features such as easy creation of scenarios and maps and allows to include the Quality of Service (QoS) parameters latency and packet loss combined with pixelation to simulate limited bandwidth. We then used this driving simulator to evaluate one of the main aspects in teleoperated driving: the latency [**VIII**]. Latency is crucial as everything the operator sees, feels and hears will be perceived with delay. Control commands, provided by the operator, are executed with delay on the remote vehicle. Altogether, both delays add up to a total delay that the remote operator or the system has to deal with. In order to be able to develop systems for teleoperated driving, it is essential to know which latencies untrained drivers are able to handle and then derive worst-case values from it. Therefore, we conducted a user study to evaluate which latencies are suitable for teleoperated driving. In the user study, participants had to drive on various road configurations with different levels of latency allowing to measure the impact of variable and fixed latency on objective driving performance and subjective perception rating.

In addition to latency, the bandwidth is another factor that needs to be addressed to achieve

[1]https://github.com/sneumeier/OpenROUTS3D

a safe remote driving experience. We analyzed bandwidth-associated requirements by conducting a further user study [**VI**]. The goal of this user study was to evaluate how far video stream can be compressed and still be sufficient for teleoperated driving scenarios. We used the obtained values to identify the minimum suitable quality of video streams in which remote control is still sufficient. This helped to identify bandwidth requirements for specific driving scenarios.

To further reduce these bandwidth requirements, it is important to use more advanced techniques [**I**]. These techniques use smarter algorithms to allow for a significant reduction of bandwidth compared to the basic approach that only applies compression as offered by the video encoder. The main idea is to distinguish between important objects and less important objects as sensed by a camera. Important parts, such as the lane in front of the vehicle or other objects, like other vehicles or pedestrians, are kept as is, while the remainder of each whole image is manipulated by a bilateral filter. This filter allows to keep important areas, but remove unnecessary details. We investigated the applicability through an online survey, in which participants were shown different driving scenarios.

Overall the contributions of this part are regarding

RQ1, conducting real-world cellular measurements with respect to teleoperated driving.

RQ2, introducing an advanced driver assistant system addressing the latency induced issues.

RQ3, a user study investigating the factor latency on driving performance and perceived workload.

RQ4, investigating possibilities to reduce the bandwidth of video streams by compression algorithms.

RQ5, design and apply advanced algorithms based on important and less important areas of the view to further reduce the bandwidth requirements of video streams.

This chapter presents our contributions published as papers. The contents of the real-world measurements are part of the paper “Measuring the Feasibility of teleoperated driving in Mobile Networks” as published in [**VII**] (Appendix page i). Speed adjustments based on latency are published in paper “Towards a Driver Support System for Teleoperated Driving” in [**III**]

(Appendix page xii). The driving simulator necessary for conducting user studies was initially introduced and described by the paper “Yet Another Driving Simulator OpenROUTS3D: The Driving Simulator for Teleoperated Driving” [**V**] (Appendix page xxii). The latency-based user study and the respective results are published in the paper “Teleoperation – The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters” [**VIII**] (Appendix page xxxi). The contents regarding visual quality are published within the paper “The Visual Quality of Teleoperated Driving Scenarios - How good is good enough?” [**VI**] (Appendix page xlviii), whereas the improvements of bandwidth requirements are published in “Data Rate Reduction for Video Streams in Teleoperated Driving” [**I**] (Appendix page lix).

The content of each publication is presented in a condensed format in the following. A more detailed version can be viewed in the appendix.

3.1 Measurements of Cellular Networks (Publication [VII])

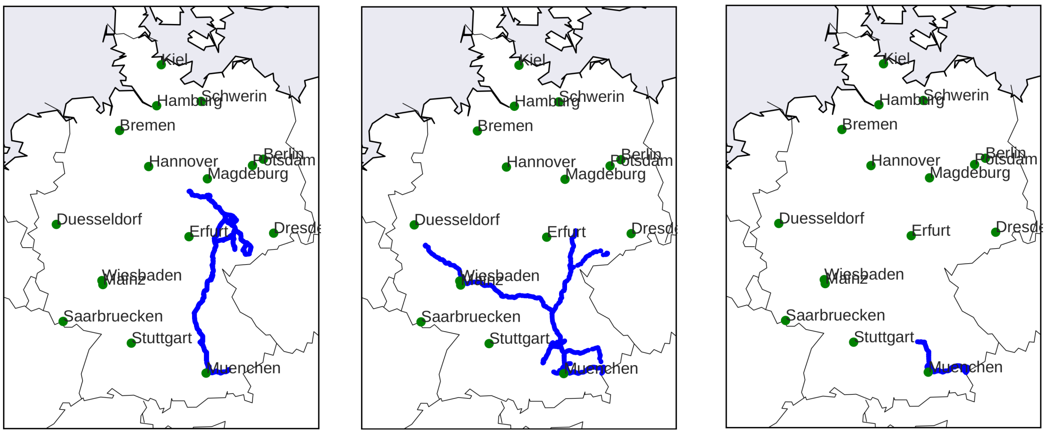
In this section, we report on our findings from measuring cellular networks in Germany with focus on remotely controlling a single vehicle.

In order to measure the performance of cellular networks, we used different hardware setups consisting of an Android-based *Lenovo B* smartphone [110] and a vehicle-mounted *SierraWireless RV50X* Long Term Evolution (LTE) gateway [159] in combination with the tools `ping`, `netradar` and `iperf3`. Measurements conducted on the smartphone consisted on the one hand of Internet Control Message Protocol (ICMP) measurements to servers in Munich and Frankfurt, based on a Android application developed by the authors [145]. On the other hand, more detailed measurements on the same platform were collected through the `netradar` [18] measurement platform, querying an endpoint server at the European Amazon Cloud.

Throughput measurements on the *SierraWireless* were executed on a car PC running `iperf3`, with an endpoint server being hosted in Munich. We performed the data collection while driving a car on different types of streets and areas in Germany, which helped to be as close a possible to real-world teleoperated driving scenarios. The measurements were carried out for 7 months in 2017. They consisted of about 78 hours and 5200 km of driving and can be split up into into 2180 km for `ping`, 2670 km for `netradar` and 354 km for *SierraWireless*, as indicated by Figure 3.1. The mobile operator was Vodafone Germany with unlimited data traffic and limitations of 100 Mbps in downlink and 50 Mbps in uplink. In order to be able to assess whether a measurement can fulfill the requirements for teleoperated driving, we defined the minimal requirements as 0.25 Mbps for downlink and about 3 Mbps for three cameras in uplink, as in [68]. Considering the latency of sensors and actuators, the upper bound for network latency was defined as 250 ms, while the jitter needs to be below 150 ms.

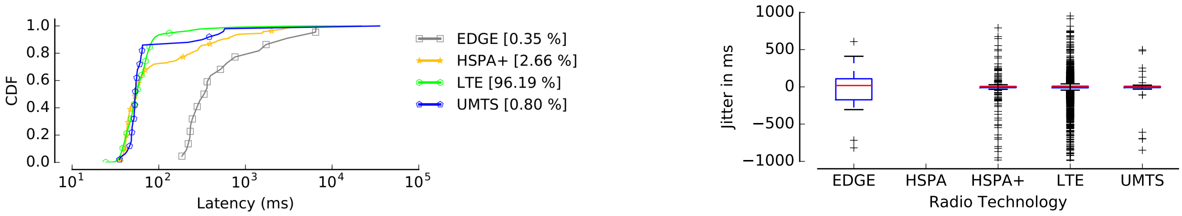
3.1.1 Results

Latency: In order to address latency (always Round Trip Time (RTT) in this paper), we conducted both User Datagram Protocol (UDP)- and ICMP- based measurements, since the latter



(a) Ping (b) Netradar (c) SierraWireless

Figure 3.1: Routes driven while measuring (a) `ping`, (b) `netradar` and (c) `iperf3`. (Source: [**VII**], ©2019 IEEE)

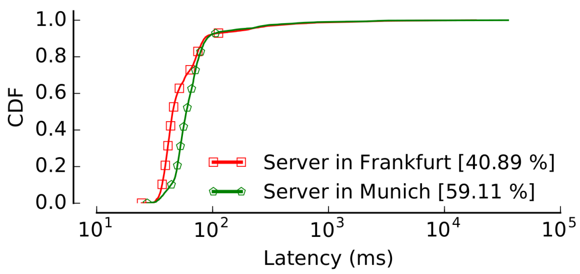


(a) CDF of the latencies. (b) Jitter of the latencies, cut at 1000ms.

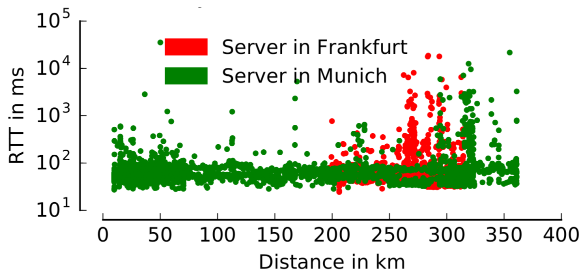
Figure 3.2: Results for latency and jitter based on the ICMP measurements. (Source: [**VII**], ©2019 IEEE))

ones might be treated differently on the way through the network [74]. The ICMP packets of the `ping` application were either sent to a server hosted in Munich (60%) or Frankfurt (40%). With a median RTT of about 55 ms (Figure 3.2a), about 96% of all measurements were below 250 ms, while the jitter with a median of about 10 ms was below the least acceptable value in about 95% (Figure 3.2b).

In order to check whether the distance between a teleoperation station and a remote vehicle has an influence on latency, we used two different server locations. It turned out that the latency for Frankfurt with about 45 ms was lower than for Munich with 59 ms, as shown in Figure 3.3a. Incorporating the average distances between measurement and server with 119 km for Munich and 270 km for Frankfurt, it could be seen that a higher distance between

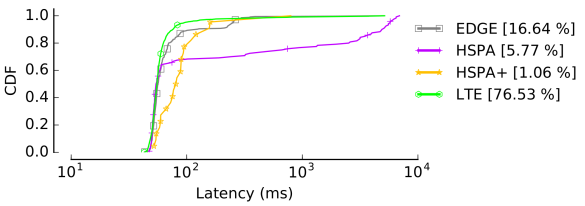


(a) CDF of `ping` latency based on the destination server in Munich/Frankfurt.

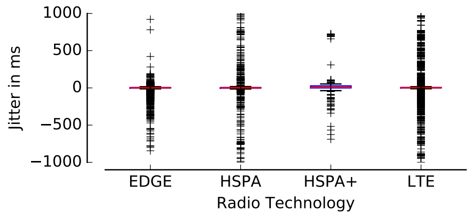


(b) Distribution of `ping` latency based on destination: Munich/Frankfurt.

Figure 3.3: Distance-related RTT values grouped by server and distance. (Source: [**VII**], ©2019 IEEE)



(a) CDF of the `netradar` latency measurements with overall median of ∼55 ms in RTT.



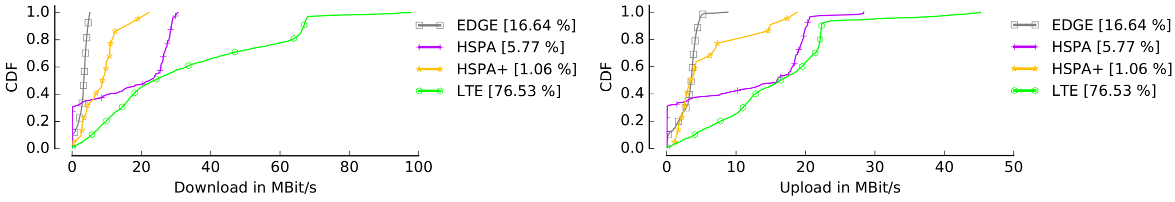
(b) Jitter of the `netradar` latency, cut at 1000 ms.

Figure 3.4: CDF of the `netradar` latency measurements combined with the results of Jitter. (Source: [**VII**], ©2019 IEEE)

teleoperation station and remote vehicle did not necessarily lead to higher latency, but the variation increased (Figure 3.3b). Because both servers were comparable in terms of their Internet connectivity, the results indicate that the distance between teleoperation station and remote vehicle can be crucial.

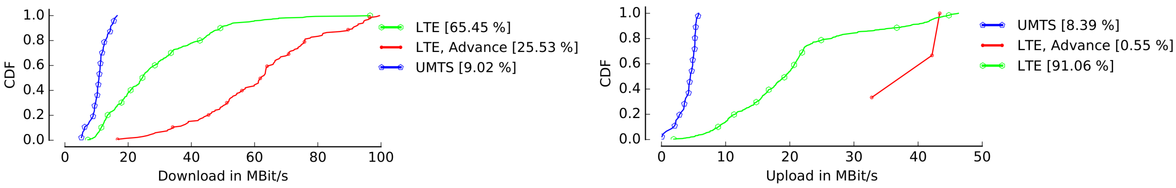
Comparing the ICMP measurements to `netradar's` UDP based latency measurements, it turned out that with an overall median of about 55 ms (Figure 3.4a), the results were comparable. It can also be seen that the latency was in about 96% of the measurements below 250 ms. For jitter, which was in about 96% of the measurements below 150 ms, the median was about 2 ms and thus more stable than for the ICMP measurements (Figure 3.4b).

Throughput: The results for the downlink measurements of `netradar` are displayed in



(a) CDF of `netradar` downlink measurements. (b) CDF of `netradar` uplink measurements.

Figure 3.5: Uplink and downlink CDFs for the `netradar` measurements. (Source: **[VII]**, ©2019 IEEE)



(a) CDF of *SierraWireless* downlink with overall median of ∼28 Mbps. (b) CDF of *SierraWireless* uplink with overall median of ∼18 Mbps.

Figure 3.6: Uplink and downlink CDFs for the *SierraWireless* measurements. (Source: **[VII]**, ©2019 IEEE)

Figure 3.5a and indicate a median of about 17 Mbps. High Speed Packet Access (HSPA) seems to be faster than HSPA+, but this is caused by the low number of measurements with those technologies. The downlink was sufficient in about 95% of the measurements and as such above 0.25 Mpbs.

Sufficient uplink speed is required to provide the operator with enough details of the vehicle's environment to allow for a safe control. As can be seen in Figure 3.5b, the median uplink speed was about 12 Mbps. In order to be sufficient for teleoperated driving, the uplink speed needs to be above the 3 Mbps threshold, which was the case in about 87% of the measurements.

For the *SierraWireless* setup, the median downlink speed was about 28 Mbps as shown in Figure 3.6a. This is nearly double the downlink speed as for the `netradar` measurements, and thus sufficient for teleoperated driving in more than 99% of the measurements. With 18 Mbps and therefore about 6 Mpbs more than for `netradar`, the median uplink speed was sufficient for teleoperated driving in about 98% of the measurements (Figure 3.6b).

Measurements on Identical Routes: Based on the previous findings we found it interesting to compare the results on identical routes in order to be able to analyze potential differences with different measurement techniques and setups. It turned out that the latency for `netradar` on those identical routes was 55 ms, while the latency for the ICMP based measurements was 57 ms. Therefore, results were roughly comparable with the same hardware and different measurement techniques. In addition, we investigated the different measurements platforms. On the identical route, the downlink speed of `netradar` was 15 Mpbs, while the one of *SierraWireless* was about 32 Mpbs and as such more than twice as high. For uplink the difference equates to 7 Mbps, as the median uplink speed of `netradar` was only 13 Mpbs, while *SierraWireless* was about 20 Mbps. This difference can be most likely attributed to the more efficient setup consisting of two antennas compared to the small antenna inside the smartphone.

Comparison of Different Scenarios: With further potential parameters influencing the network performance, it is important to investigate their potential influence on teleoperated driving scenarios. Therefore, we investigated vehicle speed, handover, signal-strength and distance to the base station in more detail.

For fast moving vehicles, it is important to see whether the vehicle speed has an influence on the network performance. Luckily, we did not observe an influence of the speed regarding latency or throughput. It may be observed that there is a better performance for the *SierraWireless* measurements at higher speeds. However, the reason for this is that these measurements were conducted on highways, which have a better coverage as already identified by previous studies [124].

Investigating the `netradar` measurements with respect to the handover, it turned out that only 12 switches in cell and 19 switches of radio technology existed. In contrast, for *SierraWireless* there were 60 cell switches during the downlink measurements and 54 during the uplink measurements, of which only one was a switch between radio technologies. Thus, this part of the measurement was not considered for further analysis regarding performance when changing radio technology. By switching cells, but keeping the same radio technology, no negative influence on latency could be seen. Unfortunately, if the radio technology changed during the measurements, the latency increased by about 15%. In case of downlink measurements in `netradar`, the cell-only switch reduced the median bandwidth by about 3 Mbps, while the results decreased by about 14 Mbps if the radio technology was also switched. For the uplink speed in `netradar`, there was no difference when switching cells while keeping the same radio technology. However, when changing radio technology the uplink decreased by about 9 Mbps. For the downlink and uplink measurements of *SierraWireless*, there was no difference when performing a handover without switching radio technology. Overall, it can be seen that teleoperated driving is feasible when switching cells. But changes in radio technology, in general, render it dangerous if the switch is not between LTE and LTE-Advanced.

We further investigated whether signal strength could influence the network parameters. Due to the mobility of the remote vehicle such changes can happen frequently. For the throughput measurements, i. e., uplink and downlink, we could see a clear tendency, indicating the better the signal strength the better the throughput. In case of latency, there was no clear tendency observable.

Finally, we investigated whether the distance to the base-station would make a difference for the network parameters. Therefore, we calculated the distances based on the data from OpenCellID [137]. With an average distance to the base station of less than 5 km, no influence on throughput (uplink/downlink) and latency could be observed.

3.1.2 Whitelisting as Possible Mitigation

In order to be able to map the results of the previous findings to real-world scenarios, it is required to avoid areas and situations where the network is not sufficient for teleoperated driving. The proposed basic approach consists of whitelisting, allowing the remote operation of vehicles only in areas that were marked as suitable for remote control. To examine such an approach, we conducted network measurements in an area covered only by LTE and LTE-Advanced. We drove a 5 km circle around the historic center of Ingolstadt with the *SierraWireless* setup. Overall the driving was repeated four times, which equates to approximately 20 km or 60 minutes in total. The driving was carried out during afternoon time to include as much commuting traffic as possible in order to put stress on the respective cellular network. We used two rounds for Transmission Control Protocol (TCP) measurements and the remaining two rounds for UDP measurements. The `ping` utility of the Linux car PC was executed continuously, i. e., one measurement step consisted of `ping`, `iperf3`

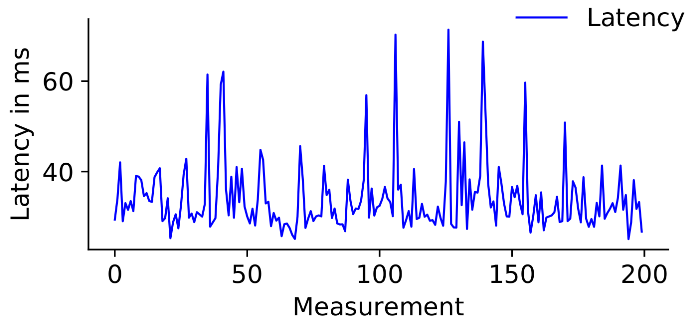
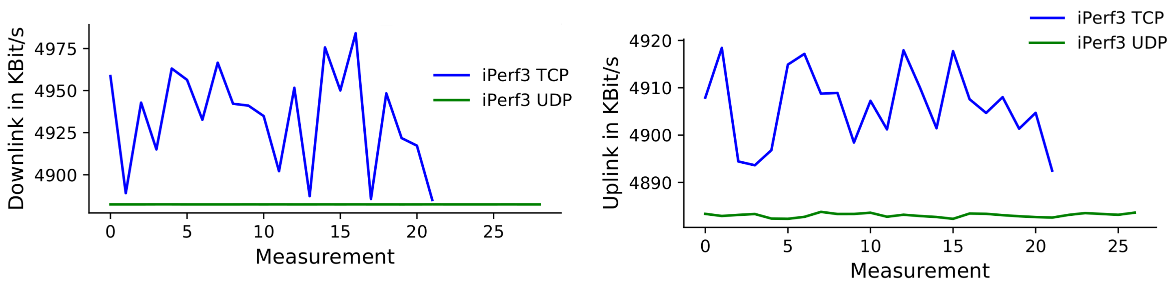


Figure 3.7: Latency measured during the test drives with median of ~31 ms, aligned by the number of measurement. (Source: [**VII**], ©2019 IEEE)

[upload|download][UDP|TCP], `ping`. To be able to see how stable the values are, we defined a maximum bandwidth of 5 Mbps for the measurements, reflecting the minimum required 3 Mbps with additional 2 Mbps as a safety margin.

As can be seen in Figure 3.7), the median latency was about 31 ms, with a maximum of 71 ms. Comparing the ICMP measurements to the RTT times of the TCP measurements, the median of 27 ms was comparable, but the maximum was about 30 ms lower than for ICMP. However, during the test drive in the whitelisted area, no measurement was above or even close to 250 ms, indicating that teleoperated driving would be possible frequently. The same applies for the jitter, which was always below the 150 ms threshold. In addition to latency, we measured uplink and downlink, specifying the introduced 5 Mbps as the target bandwidth. For TCP and UDP the downlink over time can be seen in Figure 3.8a, representing an average of about 4.94 Mbps for TCP and about 4.88 Mbps for UDP being always above the required minimum value of 0.25 Mbps. The more interesting uplink values for teleoperated driving can be seen in Figure 3.8b, indicating a median of about 4.90 Mbps for TCP and about 4.88 Mbps for UDP. These measurements were also above the required minimal value of 3 Mbps and therefore enable teleoperated driving all the time. Downlink and uplink measurements never reached the specific 5 Mbps. That is induced by the measurement method of `iperf3`, where it always tries to reach 5 Mbps from the lower bound. 16 handovers without and 5 handovers with changes in network technology happened, but as it was only between LTE and LTE-Advanced. We did see no influence on the performance. In addition, the experienced packet loss during the UDP measurements was in the order of $10^{-4}\%$ and thus not an issue for tele-



(a) Downlink

(b) Uplink

Figure 3.8: Downlink (median about 4.94 Mbps for TCP, median about 4.88 Mbps for UDP) and uplink (median about 4.90 Mbps for TCP, about 4.88 Mbps for UDP) measured during the test drives. (Source: [**VII**], © 2019 IEEE)

operated driving.

Overall it can be stated that a whitelisting-based approach is feasible and the results confirm its applicability as shown in [75].

Those measurements could only show a first tendency in the direction of teleoperated driving, as the amount of measurements, covered areas and potential measurement setups is limited. Based on those results, is important to address the latency-induced issues within teleoperated driving based. As shown in the following, this can even be achieved by using basic algorithms.

3.2 Speed Adjustments based on Latency (Publication [III])

In this section we present a methodology that adjusts the remote vehicle's speed based on latency. Reducing the remote vehicle's speed based on the latency could be a promising approach in controling a remote vehicle safely. Addressing this speed reduction, we identified three situations that require reducing the speed of the vehicle: distance to vehicles ahead, braking distance and driving through curves safely.

3.2.1 Distance to Vehicles Ahead

The first situation concerns the ability to stop accordingly to vehicles ahead, i. e., with an emergency break. In teleoperated driving the same distance rules as without latency apply, e. g., minimal safety distance of half-of-tachometer [19] as rule of thumb based on the law in Germany. We kept this rule, but extended it by latency values, to allow a distance to vehicles ahead that is safe in remote operations.

We extended the basic equation as used by [19] to cover latency, as can be seen in Equation 3.1. The elements of the equation are the speeds of the two affected vehicles (front: v_f, ego: v_e), maximal (assumed) deceleration rates (front: a_f, ego: a_e) as well as network t_l and system t_{sy} latency and reaction time t_r. Thus, it is possible to calculate the latency-induced minimal safety distance in seconds to the vehicle ahead t_s.

$$t_s = (t_l + t_{sy}) + t_r - \frac{v_f^2 - \frac{a_f \cdot v_e^2}{a_e}}{2a_f \cdot v_e} \tag{3.1}$$

3.2.2 Braking Distance

In addition to the distance to vehicles ahead, it is also important to consider the overall generic stopping distance, e. g., for unexpected obstacles. Equation 3.2, based on [161], can be used to calculate the ideal stopping distance s. Parameters are v_e for the current speed of the vehicle, a_e for the maximal deceleration rate and l_s, l_n for system and network latency, respectively.

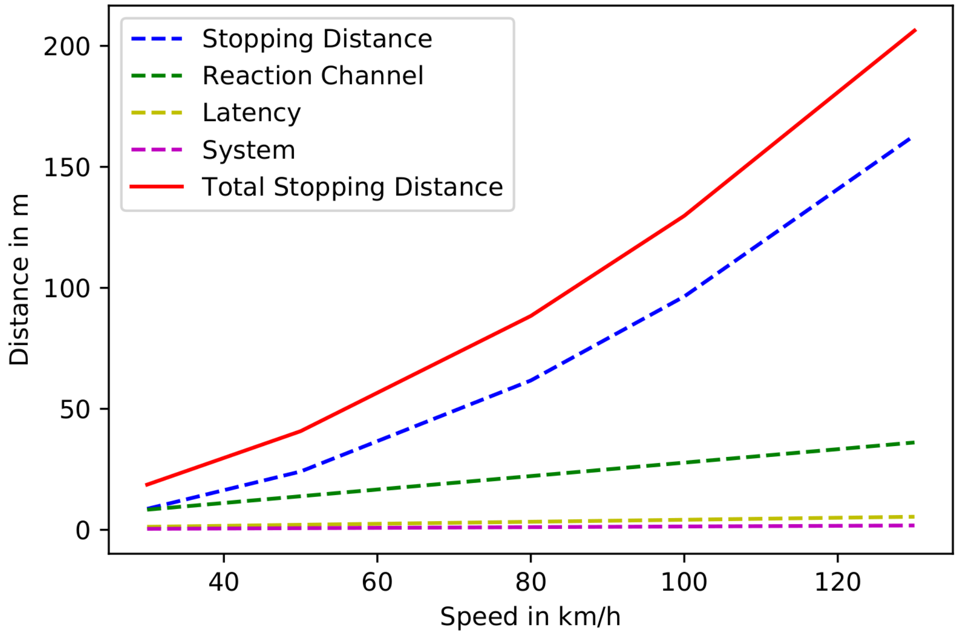


Figure 3.9: Covered distances with latencies of 150 ms for network and 50 ms for system. (Source: [**III**], ©2019 IEEE)

$$s = v_e \cdot (l_s + l_n) + \frac{v_e^2}{2 \cdot a_e} \tag{3.2}$$

Stopping distances based on speed by considering latencies can be seen in Figure 3.9, where the deceleration rate a is set to a reasonable braking of 4 m/s^2. It can be seen that the median (30 km/h – 130 km/h) stopping distance is increased by about 4.44 m considering a network and system latency of 50 ms and 150 ms, respectively (based on [156]), which plays a minor role in the overall stopping distance.

As it is not that useful for a teleoperator to only know the additional stopping distance, we calculated the adjusted speed of the remote vehicle. Given a constant deceleration, this can be expressed by Equation 3.3. Assuming that reaction time usually stays constant for remote and non-remote operators as both are humans, the adjusted speed v_l can be calculated based on system l_s and network l_n latency. In order to properly reduce the speed, the value of v_0, which indicates the speed one would drive sitting directly in the car, needs to be given, e. g., as inherited from traffic signs indicating speed limits.

$$v_l = -a * (l_s + l_n) + \sqrt{a^2 * (l_s + l_n)^2 + v_0^2} \tag{3.3}$$

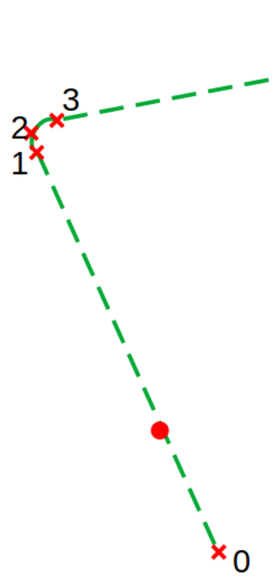


Figure 3.10: Markers of the street with the repositioned point for radius calculation. (Adjusted, Source: [**III**], ©2019 IEEE)

3.2.3 Driving Through Curves

The third major point that we considered for speed reductions are curves, as driving too fast through curves can cause slipping, i. e., the smaller the radius of the curve the lower the speed needs to be. The basic idea to address this topic is based on the fact that due to latency, all steering commands of the teleoperator will be executed by the remote vehicle with delay. It might be possible that a skilled teleoperator is able to anticipate this and reacts accordingly. However, this is based on human skills and thus potentially error-prone. The algorithm, addressing this issue, is based on the lateral acceleration a, which a vehicle will face within a curve. Therefore, it is based on the single-track model [154], as shown in Equation 3.4, where r is the curve radius.

$$a = \frac{v^2}{r} \tag{3.4}$$

In order to apply this equation, the radius of the curvature needs to be known and can be obtained from Openstreetmap (OSM) [140] data. An example of OSM data is shown in Figure 3.10, where the red crosses indicate markers of OSM, while the the green line is calculated locally, indicating the most probable road layout.

We used an outer-circle at 3 distinct points forming a triangle to approximate the curve radius based on the OSM data. We used, for example, points *(0,1,2)* in Figure 3.10, combined with the equation of Karney [94] to calculate the effective distance between these points. With Equation 3.4 and the knowledge that a usual lateral acceleration has a maximum value of approximately 0.3 m/s^2 on a dry street [125], v_c is the limiting speed for an ordinary vehicle going through that curve. When adding latency to that driving scenario, the 0.3 m/s^2 of lateral acceleration should also indicate the upper bound. If considering latency, the radius of the curve is mostly calculated as without latency, but the new first point (red dot) is repositioned based on the remote vehicle's speed and the overall latency, i. e., it indicates where the steering command would be executed if they were applied without latency. This results in a smaller radius and therefore requires reduction in vehicle speed. Figure 3.10 gives an example on how the calculation entry point is moved from the red cross 0 to the red filled circle. However, previous calculations only consider parts of the street, but for a curve all related parts are required. The example shown in Figure 3.10 thus consists of the three segments: *(0,1,2); (1,2,3); (2,3,4)*. In order to obtain the maximum speed for a curve, these parts will be calculated independently and put together to form the entire curve. The entire curve will be identified based on the different radii and the speed will be the lowest speed of all parts.

In order to be able to use the three approaches introduced above, the latency needs to be as close to an constant value as possible. This can be addressed by introducing a buffer, e. g., as shown in [32]. Usually, the buffer should be constant throughout a drive, but up to a certain frequency it can be adjusted dynamically.

3.2.4 Combination of Individual Parts

Besides the raw latency, bandwidth-related latencies and packet loss also need to be addressed. For simplification we transform them into latency-related values. For bandwidth-related video streams, this can be achieved up to a certain degree. In order to handle potential packet loss, we assumed that control and data packets are sent frequently to approximate the additional latency requirements l_l for a probabilistic packet loss.

If a teleoperator starts to control a vehicle in a real-world application, all driving relevant information such as speed limitations, curves, etc. would be displayed. The currently suitable

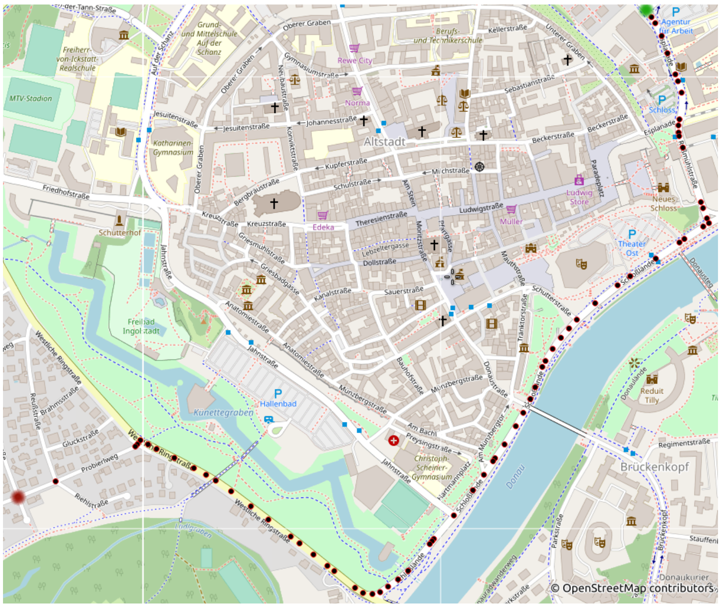


Figure 3.11: Designated route from Esplanade (green marker) to Probierlweg (red marker), annotated with OSM-points (red with black surrounding markers). (Source: [**III**], map from [140], ©2019 IEEE)

speed is based on the minimum of allowed curve speed and stopping distance. In addition it will always be checked whether the time to the vehicle ahead t_s is above the specified minimal time distance t_{min}. If this is not the case, the speed needs to be reduced by the remote operator.

3.2.5 Results on a real-world Example

In order to show whether the proposed approach is feasible in real-world scenarios, we calculated a basic example. The proposed route is in the city Ingolstadt and originates at Esplanade and ends at Probierlweg (see Figure 3.11).

Based on the latency values as measured in section 3.1.2 for the area around the historical center of Ingolstadt, realistic RTT values were 55 ms. The identified packet loss was about 0.003%. By adding 125 ms for sensors/actuators to the 55 ms RTT and 20 ms for packet loss, we considered a total latency of about 200 ms. If applying the calculation for the allowed speed based on this latency, it turned out that the speed drops to about 42 km/h, which is only 3 km/h slower as with a typical drive, assuming ideal conditions and instant changes in speed. This can be seen in Figure 3.12, where the blue and red line indicate the specific driving speeds.

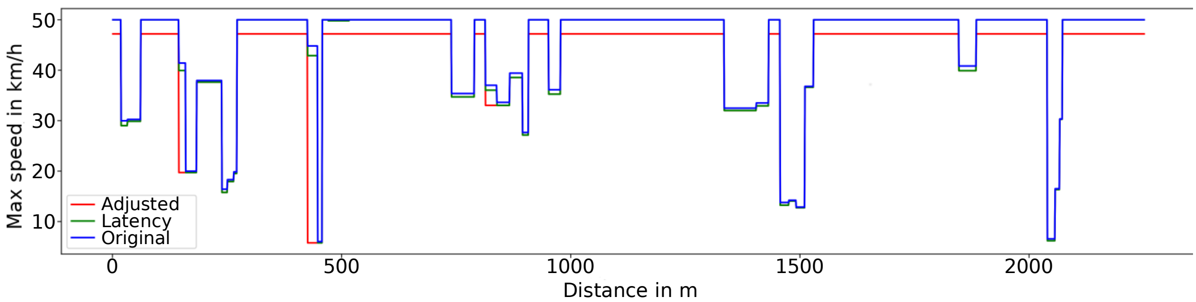


Figure 3.12: Typical driving speeds on the selected route, with blue a original speed and red as adjusted speed. Green indicates the speed adjustments without curve smoothing, treating parts of the curve as is without combining them. (Source: [**III**], ©2019 IEEE)

As shown in Section 3.5.3 or Section 3.6.4, the required bandwidth can be reduced with different algorithms. However, this may add latency, e. g., stronger compression and more advanced approaches can require more computational time [72], which can then be mapped to speed-based adjustments to counteract the additional latency.

Nevertheless, the proposed approach should be considered as a first step. It is limited by its theoretical-only approach and real-world tests are required to confirm that the approach could work.

In order to figure out, which latency can be dealt with at all, the assessment in different driving scenarios through a user study is important. We address this within Section 3.4.4, where participants had to drive various tracks with different types of latency. In order to conduct such user studies, a driving simulator is required and introduced in the following Section.

3.3 Driving Simulator OpenROUTS3D (Publication [V])

In this section we describe OpenROUTS3D, a driving simulator that we developed to address the specific needs of teleoperated driving.

The driving simulator utilizes the Unity [171] game engine and is mainly written in C#. It is licensed under GNU GPLv3 and available in github[2]. All modules that we used and developed are licensed under GPLv3 or compatible with it. Being developed with Unity, OpenROUTS3D allows the usage on all major operating systems including Windows, Linux and macOS. By using a well-known game engine and models with a sufficient level of detail, the hardware requirements are low. It can be executed on an average Intel Core i5 processor with integrated UHD Graphics as typically installed in laptop computers.

The general system architecture as shown in Figure 3.13 consists of OpenROUTS3D's internal modules, the interfaces for user interaction and third-party tools. Internal modules are responsible for tasks like vehicle physics, input/output and the generation of the environment. For obtaining real-world traffic behavior, we utilize the Traffic Control Interface (TraCI) [66] interface to exchange data with the microscopic traffic simulator Simulation of Urban MObility (SUMO) [118]. With this modular system architecture, the driving simulator is extendable. Through input files it is possible to load different maps, create user studies or add specific features based on the addon system.

Driving physics were originally based on RandomVehiclePhysics [31], but are now developed on our own, offering an easier and more maintainable way to modify the vehicle's driving behavior. So that the driving simulator can accurately take into account the network-induced influence factors for teleoperated driving, input and output is treated separately.

3.3.1 Creation of Map and Traffic

The map creation in OpenROUTS3D can happen through SUMO, Unity or a combination of both. For a quick-start, the best way is to use SUMO road networks. They are based on OSM [140] and contain information about streets, buildings, traffic lights, etc., which can be interpreted by the driving simulator. If added, the height of areas is considered and the environment is built with height information. On the left side of Figure 3.14, this type of map

[2] https://github.com/sneumeier/OpenROUTS3D

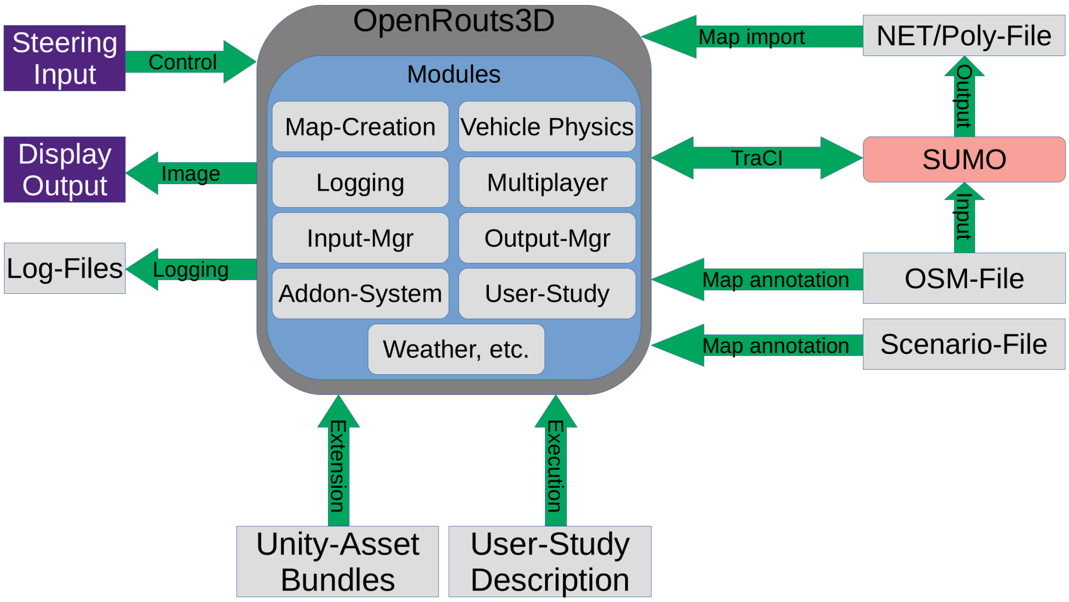


Figure 3.13: The system architecture of OpenROUTS3D. (Source: [**V**], ©2019 IEEE)

creation alongside with the manual creation of SUMO maps, e. g., with JOSM [139], can be seen.

Another way of building maps is more Unity-based and utilizes the addon system of OpenROUTS3D, where assets with an arbitrary level of detail and different objects can be included. An example of this can be seen on the right side of Figure 3.14, where we included the Windridge City-Asset [167] of Microsoft's AirSim [155] into the simulator. OpenROUTS3D provides preconfigured weather conditions that can be defined either globally or for specific areas. Based on the current weather conditions, the vehicle's driving behavior will change accordingly, e. g., longer stopping distance on snow.

It is additionally possible to extend existing environments with scenario objects. Scenario objects are meant for adding small objects or triggers to a scenario. Examples are pylons, specific latency in different areas and the starting point of the ego vehicle. Scenarios can either be built by writing specific configuration files, or by using the integrated editor.

In order to integrate traffic, the most straight forward way of obtaining traffic that behaves in a realistic way is the use of SUMO. TraCI allows to receive information about vehicle positions, traffic light states, etc., while OpenROUTS3D continuously sends the updated position of the ego vehicle, allowing SUMO to calculate artificial traffic's reaction appropriately. The

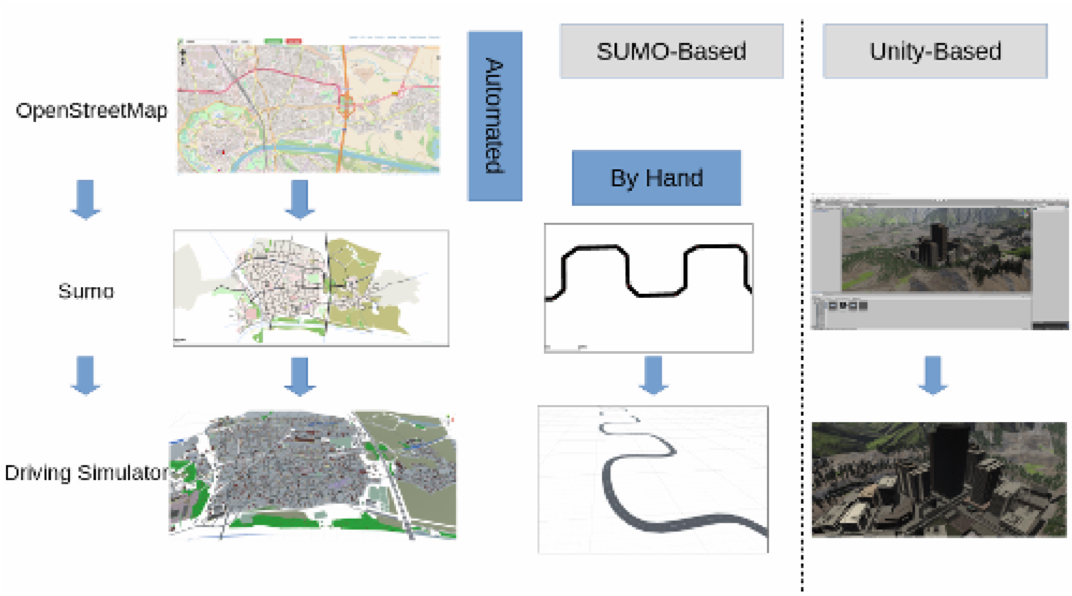


Figure 3.14: Different types for the creation of environments and maps. (Source: [**V**], ©2019 IEEE)

synchronization between the driving simulator and SUMO is handled by the driving simulator, i. e., it tells SUMO to proceed in the traffic calculation. The frequency of those updates is based on the achieved framerate of OpenROUTS3D to keep both systems in sync.

3.3.2 Input and Output

With focus on the needs of teleoperated driving, one major part of the driving simulator is distinct handling of input and output, as can be seen in Figure 3.15. The input options are manifold, supporting keyboards, gamepads and steering/pedals. For the output, different options are present. The system can be used on a typical (multi-)monitor setup by using the features of the available graphic card drivers, or for more advanced situations the use of Head-Mounted Display (HMD) is possible.

For simulating the network-induced issues of teleoperated driving, the main feature is the ability to configure latency independent for input and output, which we realized through buffers. We store every input command and every generated display image in a distinct buffer and therefore are able to further manipulate those entries. We chose this differentiation between input and output to allow for studying different behaviors, e. g., in real-world appli-

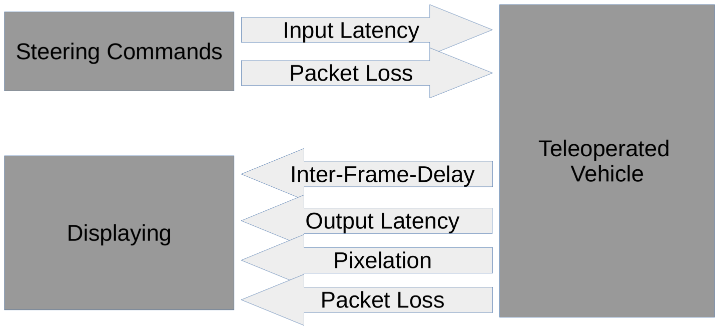


Figure 3.15: Overview of the Input/Output parameters. (Source: [**V**], ©2019 IEEE)

cations the communication to the vehicle may have a different performance than on the way back, which might have different effects. By delaying the final delivery of input commands or displaying the generated image, latency can be simulated efficiently. With the pixelation, the stream quality itself can be manipulated by adjusting its quality and frame-rate. We achieve this by applying pre-rendering of the image and manipulating it when required. The configuration of inter-frame delay drops images from the buffer. This allows to adjust the framerate and simulate different real-world cameras. Based on a probabilistic model it is additionally possible to drop random entries of input and output buffers to simulate packet loss.

3.3.3 Additional Features

In addition to the basic set of features required for teleoperated driving, the driving simulator has further features that can be enabled through the addon system for advanced functionality.

Logging and Replay Feature: Basically, OpenROUTS3D writes different log files to allow a detailed analysis of drivings afterwards. Typically, multiple log files exist and consist of drive-logs with detailed frame-based positions, special collision logs and the results of potential questionnaires. In addition to the creation of log files, it is possible to stream the drive logs to another system in real-time. This can be helpful in situations in which a drive should be evaluated in real-time, e. g., in a user study where the participant sits in a different room than the supervisor, but the supervisor needs to know what is going on to intervene if required. Based on the detailed logging functionality, we implemented a replay system. This system reads the stored contents of the log files combined with the scenario and allows watching the drive from different arbitrary perspectives. Physics are disabled then, which means that updates of position, etc. only happen based on the log files and thus allows for a real replay. This has the huge advantage that the replay speed can be adjusted arbitrarily, based on interpolation. We intended this feature for analyzing unclear situations in a recorded drive during a user study. One side effect of the replay mode is, that it can be used for the application of algorithms that are slower than real-time. A drive can be conducted normally and the algorithm will be executed on the replay. This allows for a smooth and uninterrupted drive first, and the application of whatever algorithm afterwards without being stuck to time-constraints.

Multiplayer: In order to be able to simulate more complex scenarios with more than one human participant, we implemented a multiplayer mode. This is especially useful for investigating the behavior of operators and normal drivers in combination. The behavior of a teleoperated vehicle may differ from other vehicles and the reaction of the others to it can be investigated. We implemented this by considering one instance of OpenROUTS3D as host, while the others join as clients. The host is also responsible for handling SUMO, which means that it will collect the position of all player vehicles and transfers them to SUMO. Afterwards, the host gathers the updated information of non-player-vehicles from SUMO and sends position updates to all clients. Managing the traffic updates this way keeps the requirements on the clients as small as possible. To join a multiplayer session the only need to have the same map available.

User Study Mode: Conducting user studies is an important aspect in the research domain of teleoperated driving. Therefore, OpenROUTS3D offers the ability to easily set up and conduct user studies. We put our main focus on the aspect that a participant should be interrupted by the supervisor as infrequently as possible. We achieved this by providing a study procedure that will run autonomously, lowering the communication and distraction effort significantly. The study description containing all explanations, driving tasks and related questionnaire is prepared prior to a participant's drive. The description of such a study happens with XML-files defining individual inputs such as single-select, multi-select, slider-input, free-text, etc. to build a questionnaire. Thus, user studies as shown in Figure 3.16 can be created easily by hand or automated with a tool.

In order to extend or adjust the behavior of OpenROUTS3D, we implemented an **addon system**. It allows the user to add new or replace existing objects in the driving simulator, by simply dropping a properly packed bundle into the addon folder. The specific structure

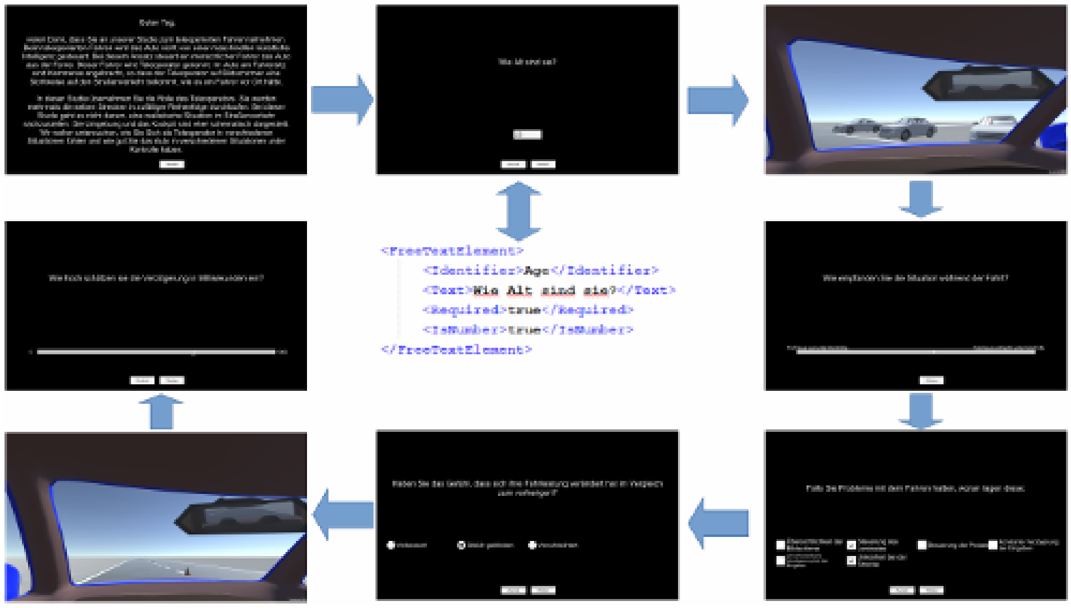


Figure 3.16: Screenshots of an exemplary user study and a xml-based question within OpenROUTS3D. (Source: [**V**], ©2019 IEEE)

consists of an Unity asset bundle and an icon which will be displayed in the main menu of OpenROUTS3D. Within the main menu, addons can be activated or deactivated. We implemented those addons in a way, in which they do not contain any information about the scene they are loaded into. This allows the development of generic addons. They are registered as an *anchor* when loading the simulation. We provide a basic set of widespread addons with OpenROUTS3D, e. g., the replay functionality and the network logging are realized as addons. Further addons provide basic features such as displaying a timer, loading textures based on OSM information or freeze the point of view to avoid arbitrary camera movements.

Although OpenROUTS3D is fit for the needs of teleoperated driving, it is yet in an initial stable version and needs to be improved further to be more accurate in simulating real-world behavior.

Utilizing this driving simulator allows to conduct user studies with respect to teleoperated driving. Focusing on the effects of latency, OpenROUTS3D is utilized in the user study presented in the next Section.

3.4 User Study on the Impact of Latency (Publication [VIII])

In this section we cover the influence of latency, one of the main aspects in teleoperated driving. Latency is critical as everything the teleoperator sees is already out of date and the environment may have changed in the meantime. The same applies vice versa. The control commands issued by the teleoperator are executed with a delay and may not longer match to the current driving situation. Overall, both delays add up to a total delay the remote operator and the system have to deal with.

In order to investigate the influence of latency on the perceived (subjective) workload and on the driving performance, we conducted an on-site user study, where participants were using the OpenROUTS3D driving simulator. Thus, we utilized objective (driving performance) and subjective measures (NASA Task Load Index (NASA-TLX) questionnaire, NASA-TLX [76] and a self-defined questionnaire) during the drive. Selected scenarios were derived from real-world applications and reflect parking as well as general vehicle handling. We used a three-monitor setup with a pedal and a steering wheel for input and a car-like gaming seat. The setup of this user interface is comparable to other setups as shown by *Designated Driver* [39] and *Phantom Auto* [75]. These two companies develop teleoperated driving systems and already test them on real streets.

Our driving scenarios selected for the study are shown in Figure 3.17. The participants had to follow the road from their starting position to its end, which on the figure (images 1 to 3) corresponds to a drive from right to left. An exception here is the last scenario (4th image in Figure 3.17), in which the participants had to approach the parked vehicles and park between them. We chose daytime and sunny weather for the user study so that the environmental impact on the results was kept to a minimum.

In order to fulfill the driving task, the participants had to drive every scenario with constant artificial one-way latencies: 0.0 s (**no**), 0.15 s (**small**) or 0.3 s (**large**) each for input and output (i. e., RTT of 0.3 s or 0.6 s). In addition to the constant latency, we added a scenario with **varying** artificial latency. We configured different sections of the road to have a different latency. These section are shown in Figure 3.18. The one-way latency was 0.0 s, 0.05 s, 0.075 s, 0.15 s and 0.25 s each for input and output (i. e., RTT of 0.1 s, 0.15 s, 0.3 s and 0.5 s).

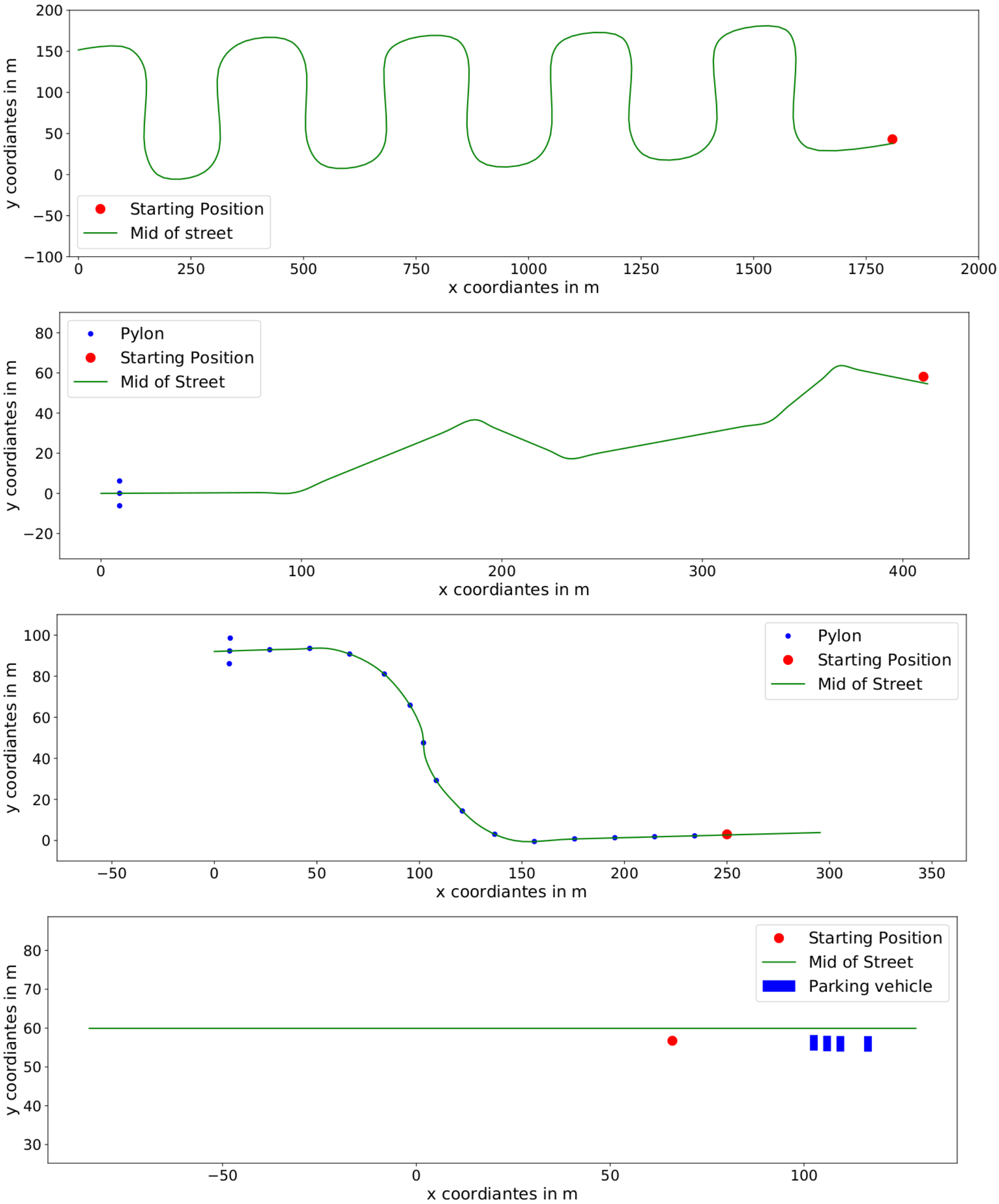


Figure 3.17: The user study's maps driven with constant latency, from top to bottom: Long-Track/Practice, Snake, Pylon, Parking. (Source: [**VIII**])

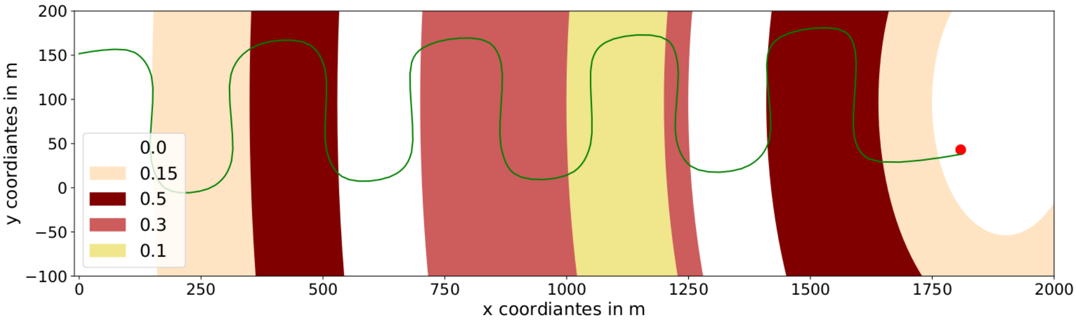


Figure 3.18: Different latencies indicated by different colors in the LongTrack scenario. (Source: [**VIII**])

Besides the artificially inserted latency there was also latency introduced by the setup, i. e., the time span between input and the display of this input on the monitor. We measured the signal delay between the keystroke and the final display on the monitor using a button and a photodiode and found a median delay of 67 ms. This means that the total experienced latency for the participants consisted not only of artificial latency but also of system latency, e. g., 0.367 s for the **small** latency.

We conducted the user study itself with a within-subjects experimental design, where each participant had to drive each scenario multiple times with different latencies. In total each participant drove 10 scenarios, apart from the practice drive to get familiar with the vehicle's physics and the steering.

At the beginning participants were introduced to the study and the cockpit was adjusted to fit their size, i. e., steering wheel, pedals and seat were positioned accordingly. The study at the driving simulator started with an explanation of teleoperated driving and a short introduction on how to control the vehicle. Afterwards, the participants were able to practice. Subsequently, after answering some basic *pre-driving* questions, the four relevant driving blocks started. At the end of each block, consisting either of one fixed latency with all maps or the variable latency-based `LongTrack`, the participants had to answer *post-latency* questions. Subsequent to the last driving block, the questionnaire contained some *post-driving* questions.

In general, each drive was analyzed with focus on objective and subjective values. The objective measurements were analyzed based on the drive-log and consisted of lateral deviation from the center line as the *Mean Lateral Position (MLP)* and the *Standard Deviation of*

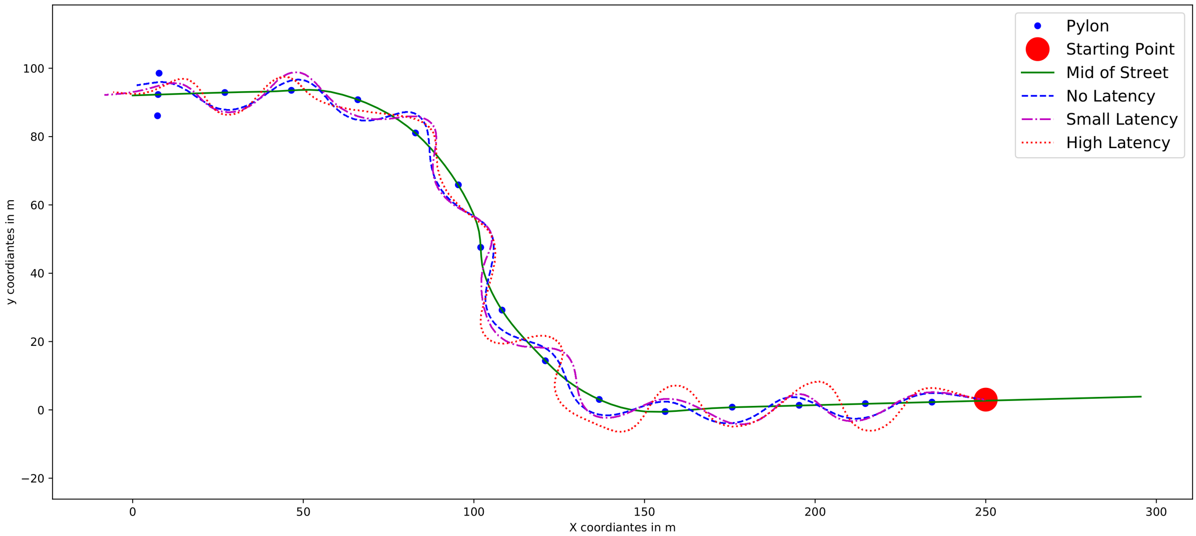


Figure 3.19: Comparison of different drives of one participant on the `Pylon` scenario. (Source: [**VIII**])

Lateral Position (SDLP) [98]), *maximum steering angle*, *out of lane ratio*, *average speed* and *acceleration* as well as *deceleration*. Altogether, we have chosen these values because they help to analyze a drive in terms of an accurate and smooth drive. This was important in order to determine the controllability of the vehicle under the given circumstances. The perceived impact of latency on the participants was measured with the NASA-TLX, [76] and a self-defined questionnaire – written in German, 5-point Likert scale –, which allowed to analyze the perceived (subjective) workload, get information on the participants stress and identify the cause of potential issues.

In order to discuss the results properly, we will at first briefly explain the results of the objective (Subsection 3.4.2) and the subjective measurements (Subsection 3.4.3). Afterwards we discuss (Subsection 3.4.4) the objective impact of constant latency and perform a comparison of variable and constant latency. Finally, we discuss the influence of latency has on the perceived workload

3.4.1 Results

A total of 28 subjects participated in the study, six of whom were female. All were required to have a driver's license; their median age was 26.89 years.

	Pylon Mdn (IQR)			**Snake Mdn (IQR)**			**Parking Mdn (IQR)**			**Long Track Mdn (IQR)**
Latency	**no**	**small**	**large**	**no**	**small**	**large**	**no**	**small**	**large**	**varying**
Lane Keeping Performance										
MLP	1.85 (.83)	2.13 (.55)	2.21 (.87)	2.52 (.73)	2.73 (.89)	2.66 (.52)	-	-	-	2.94 (.54)
SDLP	1.07 (.40)	1.27 (.45)	1.41 (.54)	.96 (.36)	1.07 (.58)	1.30 (1.02)	-	-	-	1.59 (.78)
Max. Steer Ang.	23.80 (8.10)	27.95 (8.48)	35.00 (6.07)	13.30 (4.55)	14.15 (5.90)	16.75 (13.92)	35.00 (6.2)	35.00 (5.65)	35.00 (8.10)	15.55 (7.48)
Out of Lane Ratio	0 (0)	0 (.03)	.03 (.08)	0 (.01)	0 (.03)	.02 (.08)	-	-	-	.027 (.05)
Speed and Acceleration/Deceleration										
Avg. Speed km/h	20.07 (5.49)	17.93 (8.60)	14.94 (6.18)	38.31 (13.92)	34.56 (13.38)	31.39 (9.66)	4.29 (3.31)	4.57 (4.46)	4.00 (3.03)	53.28 (3.31)
Mdn. Acc m/s^2	1.13 (.97)	1.15 (.97)	1.28 (.72)	1.23 (.62)	.96 (.90)	1.62 (1.11)	.80 (.36)	.85 (.66)	.89 (.40)	1.75 (.50)
Mdn. Decc m/s^2	3.05 (4.30)	2.84 (3.95)	2.14 (4.27)	1.53 (4.28)	5.82 (3.77)	4.38 (2.63)	1.36 (1.03)	1.76 (1.33)	1.64 (1.30)	4.99 (1.18)

Table 3.1: Results of the driving performance measures in medians combined with the Interquartile Range (IQR). (Source: [**VIII**])

3.4.2 Driving Performance

For the description of the objective results, we will talk about the significant differences between the latencies of each scenario. The actual results can be seen in Table 3.1. Figure 3.19 shows the latency-based results of one participant on the `Pylon` track, where different latencies led to different tracks on the scenario.

Pylon Scenario: For the *MLP*, no significant difference between any of the latencies could be detected. The *SDLP* in contrast revealed a significant difference between **no** and **large** latency. For the *maximum steering angle*, there was a significant difference in the pair-wise comparison of all latencies. In the *out of lane ratio*, participants left the lane significantly more often in the **large** condition compared to the lower latencies. For *average speed*, a significant difference could be seen between **no** and **small** latency as well as **no** and **large** latency, whilst

	No Mdn (IQR)	**Small Mdn (IQR)**	**Large Mdn (IQR)**	**Varying Mdn (IQR)**
Effort	1.00 (1.00)	2.00 (2.00)	3.00 (1.00)	2.00 (1.00)
Frustration	1.00 (1.00)	1.00 (3.00)	3.00 (1.00)	2.00 (3.00)
Performance	1.00 (2.00)	2.00 (2.00)	3.00 (3.00)	2.00 (1.00)
Mental Demand	1.00 (2.00)	1.00 (2.00)	3.00 (2.00)	3.00 (1.00)
Physical Demand	0.00 (1.00)	1.00 (2.00)	2.00 (3.00)	1.00 (2.00)
Temporal Demand	0.50 (1.00)	1.00 (2.00)	1.00 (2.00)	1.00 (2.00)
Overall Workload	4.50 (5.75)	7.50 (6.00)	13.00 (8.50)	10.00 (5.75)

Table 3.2: Medians and Interquartile Range (IQR) from the NASA-TLX questionnaire. (Source: [**VIII**])

no significant difference for *acceleration/deceleration* could be detected.

Snake Scenario: Like in the `Pylon` scenario, no significant differences could be identified for *MLP*, but *SDLP* revealed a significant difference between **no** and **large** latency. The *maximum steering angle* was significantly higher for the **large** latency compared to the **no** latency condition. For the *out of lane ratio* no significant difference could be identified. In case of *average speed*, significant differences were found between **no** and **large** latency and **small** and **large** latency. A significant difference for *acceleration* was present between **no** and **large** latency. For *deceleration*, both **small**, and **large** were significantly different from **no** latency.

Parking Scenario In the `Parking` scenario, no significant difference could be identified. However, *lane keeping performance* was ignored, as parking always happened next to the street and the *maximum steering angle* was also maxed under all conditions.

LongTrack Scenario As this scenario was only conducted once with different latencies at specific sections of the scenario, a comparison between latencies was not possible. However, since we wanted to do a comparison between constant and **varying** latency, we compared it to the `Snake` scenario. We considered the two different types of road topology and thus only used *MLP* and *SDLP*, as both are widely used metrics to evaluate driving errors [98]. For *MLP* all constant latencies were significantly lower than the values obtained from the **varying** latency scenario. With *SDLP*, a significant difference existed only in comparison to the **no** latency condition.

3.4.3 Questionnaire

In addition to the objective results, we also did an analysis on the perceived (subjective) workload. The results of the NASA-TLX can be seen in Table 3.2 and will be briefly described in the following.

Considering the *overall workload* of the NASA-TLX, results showed a significant difference between all constant latencies. For the **varying** latency a significant difference only existed when comparing it to **no** latency. By comparing the sub-scales of the NASA-TLX, significant differences could be seen in all of them. For *frustration*, *performance* and *mental demand*, a significant difference was revealed between **no** and **large** latency and between **varying** and **no** latency. In addition, *frustration* significantly differed between **small** and **larg** latency and between **small** and **varying** latency. *Mental workload* additionally differed significantly between **small** an **varying** latency. The only significant difference in *physical demand* could be seen between **no** and **large** latency. Analyzing the *perceived temporal demand* did not reveal any significant differences.

3.4.4 Discussion

Based on the results of the user study, we will now discuss the results with respect to: impact of constant latency, constant latency compared to varying latency and influence of latency on perceived workload.

Impact of Constant Latency: We can summarize that the driving performance decreased with higher latency. However, this was strongly dependent on the complexity and driving speed of the scenarios. For scenarios with a lot of steering inputs and higher speed like the `Pylon` one, even the **small** latency could be too high to allow for a safe drive. For scenarios with a lower complexity but greater speed like in the `Snake` scenario, it turned out that driving with **small** latency revealed similar performance as with **no** latency. The impact of **large** latency on this scenario showed a decrease in the driving performance. In contrast, for slow driving scenarios the **large** latency might be acceptable, as no real difference between the different latencies could be noticed. Taking all this into account, we consider a RTT below 300 ms as being sufficient for skilled and trained teleoperators. This is comparable to the results of latency analysis for video games [43].

Constant vs. Varying Latency: In addition to the impact of constant latency, we also compared the effects of **varying** latency and constant latency. By considering the different road configurations that were driven with constant and **varying** latency, it only made sense to compare `LongTrack` and `Snake`, as they have a similar type of road configuration. Due to the two different road configurations, the best marker for comparison was the SDLP, as this can be interpreted as a deviation from the ideal driving lane represented by the MLP [99] and thus was more accurate for the given constellation. Using SDLP as the most suitable metric, differences could only be seen between **small** and **varying** latency. Therefore the results of Kang et al. [93] and others, who state that constant latency leads to better driving performance than **varying**, could not be confirmed by us with this comparison. However, the difference in the road configurations could have a huge impact on this findings and thus the result needs to be treated carefully.

Perceived Workload: With the NASA-TLX, we selected an indicator that allows us to see how demanding driving tasks were. As expected, there was a perceived difference between **no** and **large** latencyand **no** and **varying** latency, which showed that the perceived workload increased. However, the overall workload indicated that there is already a difference between **no** and **small** latency. That means that for applying teleoperated driving in real-world scenarios, a skilled and specially trained operator seems indispensable. We also revealed that there was a significant increase of perceived workload between the **small** and **large** latency. This means that the reduction of latency should be one of the main goals in teleoperated driving.

Although the user study revealed interesting results, we only used a limited set of latencies, participants and scenarios. Thus, results can be considered as first tendencies, but further more complex studies need to be carried out.

Besides addressing the impact of latency, it is also important to address the other limiting factor in cellular networks, the bandwidth. This is addressed in the following Chapter 3.5.3 and further improved in Chapter 3.6.4.

3.5 Video Encoding for Bandwidth Reduction (Publication [VI])

In this section we address bandwidth as one limiting factor in the use of teleoperated driving. Our main idea consists of reducing the bandwidth requirements of the video stream efficiently by applying video codec based compression. However, the quality cannot be reduced arbitrarily, as a minimal level of quality is required to allow the teleoperator to observe the vehicle's environment. Therefore, we conducted an online survey in which participants had to rate the visual quality of real-world driving scenarios based on the different compression parameters. The participants had to rank a total of ten compressed videos based on the perceived video quality and the respective usability for teleoperated driving.

3.5.1 Preparation of Video Clips

Before conducting the user study, it was important to find and prepare appropriate video clips, reflecting different weather and light conditions [100]. We did this by comparing different available datasets based on image quality, scenario diversity and image frequency: KITTI [62], Lyft [95], Waymo [175], A2D2 [67] and Udacity [169]. The Waymo dataset, of which only the front camera was used, consists of 1,000 driving segments, with 20 seconds of length recorded at 10 Hz [116] and thus offered the best overall package. We visually analyzed the video clips offered by Waymo with respect to their usability for the study. The selection happened based on two factors regarding the complexity of a driving situation and the diversity of weather and light conditions. Finally, we chose ten video clips (Figure 3.20) representing different scenarios that consist of different driving situations and various weather and light conditions.

We carefully cropped these ten video clips to full-hd resolution, which is usually a best-case resolution for teleoperated driving [**VII**]. Unfortunately, the dataset was recorded at 10 Hz, which led to a slow and inert display of scenarios. After testing and reviewing multiple interpolation settings with five pre-study participants, we decided to use a 20 Hz framerate without interpolation. This setting provided the lowest influence on the perception, i. e., it avoided the introduction of interpolation artifacts. In order to achieve different video qualities,

(a) Scene 0 (b) Scene 1 (c) Scene 2

(d) Scene 3 (e) Scene 4 (f) Scene 5

(g) Scene 6 (h) Scene 7 (i) Scene 8

(j) Scene 9

Figure 3.20: Scenario pool of the user study. (Based on: [175], Source: [**VI**], ©2020 IEEE)

several codec parameters can be adjusted. For this work, we altered the parameters `preset`, `tune`, `bitrate`, `codec`, `Constant Rate Factor (CRF)` and `resolution`, which resulted in a total of 25,920 video clips generated with FFMpeg [58].

During the generation of the video clips, they were rated based on Netflix's VMAF score [112], to allow a first assessment of the potentially perceived video quality. Because not all of the 25,920 video clips could be rated by the participants, we clustered every scenario (Figure 3.20) into five quality levels based on the calculated VMAF value using the KMeans algorithm. Finally, from every cluster, we chose the video clip closest to the cluster center for

	Parameters						Parameters				
Scenario	**Type**	**#**	**Cluster**	**VMAF**	**Bitrate**	**Scenario**	**Type**	**#**	**Cluster**	**VMAF**	**Bitrate**
0	Highway Sunny Day	1760	0	21.56	202.73	5	Suburban Rainy Day	1549	0	22.84	197.22
			1	37.01	401.56				1	39.73	392.68
			2	55.52	1580.36				2	58.90	1566.02
			3	73.53	3204.66				3	75.48	3174.58
			4	89.51	6224.72				4	89.61	6156.24
1	Highway Rainy Day	1650	0	23.16	189.61	6	Urban Rainy Day	1623	0	21.28	206.85
			1	36.77	402.28				1	36.61	402.08
			2	52.56	1588.01				2	54.37	1589.08
			3	69.81	3188.80				3	72.65	3228.56
			4	86.88	6295.02				4	89.20	6224.58
2	Suburban Sunny Day	1640	0	22.52	210.12	7	Rural Sunny Day	1551	0	22.60	207.23
			1	39.50	398.34				1	39.58	407.40
			2	59.08	1567.11				2	57.35	1584.98
			3	77.59	3272.64				3	73.99	3282.60
			4	91.92	6228.10				4	89.05	6397.42
3	Suburban Rainy Night	2038	0	22.62	237.50	8	Urban Sunny Day	1771	0	20.23	175.76
			1	36.87	414.80				1	33.13	346.72
			2	52.52	1631.63				2	48.45	1414.21
			3	69.17	3294.43				3	66.39	3016.14
			4	86.25	6397.35				4	88.21	5875.69
4	Highway Rainy Dawn/Dusk	2133	0	25.17	220.46	9	Urban Sunny Night	1891	0	22.28	177.00
			1	41.24	413.37				1	39.31	371.14
			2	56.46	1582.23				2	56.99	1555.78
			3	72.14	3168.80				3	73.77	3154.67
			4	87.98	6124.03				4	88.58	6133.62

Table 3.3: Parameters of the selected scenarios for each of the groups (# of video clips with Video Multi-Method Assessment Fusion (VMAF) above 15). (Source: [**VI**], ©2020 IEEE)

presentation during the survey. In order to avoid quality levels that were obviously too bad, we only used videos with a VMAF rating above 15. We also calculated other metrics such as Multiscale SSIM (MS-SSIM), but results supported the claim of Li et al. [112] that they are not accurate enough. The finally selected video clips combined with a short overview of important values can be seen in Table 3.3.

3.5.2 User Study

We conducted the user study online using SoSci Survey [109], where participants could use their own devices to participate. At the beginning of the survey, teleoperated driving was introduced to the participants, followed by a questionnaire about demographic data.

Each participant had to rate ten randomly chosen video clips, one at a time. The rating consisted of the perceived video quality (Mean Opinion Score (MOS), `1|Bad` to `5|Excellent`) and an assessment whether they think that the displayed video quality would be sufficient for remotely driving a vehicle on their own or feeling safe while being remotely driving by an expert. The options on a four point Likert scale were: `yes, rather yes, rather no, no`. The instructions of every video stated that video clips should be watched in fullscreen mode awaiting potential buffering before playing.

The design of the survey made it possible for participants to use any type of device, which is not ideal for video ratings, e. g., smartphones with small displays and high resolutions will lead to different results than typical computer monitors. Although every participant was told not to use handheld devices, we recorded supporting data such as resolution, deviceToPixelRatio, browser and operating system for every participant. This allowed us to filter unsuitable setups during the analysis process.

3.5.3 Results

In total 95 out of 115 (82%) participants finished the questionnaire properly and could be considered for further analysis. After we removed resolutions that were too low (below 1600x900) or too high (above 2000x1250) considering the deviceToPixelRation – both which might have a strong negative influence on the perceived quality – a total of 70 valid participants remained. This left us with at least 10 user ratings per video clip. The survey period covered about one month, starting in January 2020. The age of the participants was about 30 years, 80% of them were male and 20% were female.

The most interesting results as can be seen in Table 3.4 are in the column *Controlling*. This column indicates participants' rating on whether they would trust themselves to remotely control a vehicle with the perceived quality, i. e., which video quality is rated barely sufficient. For all scenarios, except scenarios 3 and 4, there is at least one quality level that was rated

S	Q	#	V	**Controlling**		**Passenger**		**Rating**		S	#	V	**Controlling**		**Passenger**		**Rating**	
				AVG	MED	AVG	MED	AVG	MED				AVG	MED	AVG	MED	AVG	MED
	0	15	21.56	1.40	1.00	1.40	1.00	1.33	1.00		15	22.84	1.13	1.00	1.07	1.00	1.13	1.00
	1	16	37.01	2.38	2.00	2.12	2.00	2.38	2.50		14	39.73	1.64	1.50	1.64	1.50	1.64	2.00
0	2	10	55.52	2.90	3.00	2.80	3.00	3.70	4.00	5	14	58.90	2.64	3.00	2.71	3.00	3.21	3.00
	3	17	73.53	3.00	3.00	2.71	3.00	3.53	4.00		14	75.48	3.14	3.00	2.86	3.00	3.86	4.00
	4	12	89.51	3.17	3.00	2.92	3.00	4.17	4.00		16	89.61	3.12	3.00	3.06	3.00	3.88	4.00
	0	19	23.16	1.53	1.00	1.42	1.00	1.53	1.00		11	21.28	1.09	1.00	1.09	1.00	1.27	1.00
	1	14	36.77	1.50	1.00	1.50	1.00	1.57	1.50		11	36.61	2.18	2.00	2.09	2.00	2.91	3.00
1	2	14	52.56	1.86	2.00	1.43	1.00	2.00	2.00	6	16	54.37	2.69	3.00	2.38	2.50	3.00	3.00
	3	15	69.81	2.60	3.00	2.67	3.00	3.07	3.00		13	72.65	2.77	3.00	2.46	2.00	3.23	3.00
	4	10	86.88	3.00	3.00	3.00	3.00	4.10	4.00		10	89.20	2.50	2.50	2.50	2.00	3.60	4.00
	0	17	22.52	1.53	1.00	1.59	1.00	1.76	2.00		15	22.60	1.47	1.00	1.47	1.00	1.60	2.00
	1	17	39.50	2.12	2.00	2.00	2.00	2.59	2.00		15	39.58	2.27	2.00	1.73	2.00	2.00	2.00
2	2	13	59.08	2.92	3.00	3.08	3.00	3.62	4.00	7	16	57.35	2.69	3.00	2.69	3.00	3.44	3.00
	3	15	77.59	3.00	3.00	2.73	3.00	3.53	4.00		11	73.99	2.73	3.00	2.18	2.00	3.82	4.00
	4	13	91.92	3.15	4.00	3.00	3.00	4.00	4.00		17	89.05	2.76	3.00	2.94	3.00	4.29	4.00
	0	11	22.62	1.00	1.00	1.09	1.00	1.00	1.00		14	20.23	1.57	1.50	1.57	1.00	1.86	1.50
	1	17	36.87	1.35	1.00	1.35	1.00	1.47	1.00		13	33.13	2.00	2.00	2.00	2.00	2.08	2.00
3	2	12	52.52	1.25	1.00	1.25	1.00	1.33	1.00	8	14	48.45	2.21	2.50	2.21	2.00	2.71	2.50
	3	17	69.17	1.53	1.00	1.59	1.00	1.76	2.00		14	66.39	2.64	3.00	2.64	3.00	3.36	3.00
	4	11	86.25	1.82	2.00	1.64	1.00	2.09	2.00		16	88.21	2.62	3.00	2.88	3.00	3.81	4.00
	0	13	25.17	1.08	1.00	1.00	1.00	1.00	1.00		14	22.28	1.71	2.00	1.57	1.00	2.00	2.00
	1	16	41.24	1.06	1.00	1.06	1.00	1.06	1.00		15	39.31	2.27	2.00	2.07	2.00	2.27	2.00
4	2	15	56.46	1.07	1.00	1.00	1.00	1.20	1.00	9	11	56.99	2.36	3.00	2.09	2.00	3.09	3.00
	3	12	72.14	1.42	1.00	1.42	1.00	1.58	2.00		13	73.77	2.92	3.00	3.00	3.00	3.31	4.00
	4	15	87.98	1.67	1.00	1.60	1.00	2.07	2.00		12	88.58	2.50	2.00	2.33	2.00	3.33	3.00

Table 3.4: Results (**#** of ratings) of the survey, with green areas that indicate videos participants would rather trust as driver and orange areas that indicate videos participants would trust as passenger, for Scenarios (**S**), Qualities (**Q**) and VMAF (**V**). (Source: [**VI**], ©2020 IEEE)

sufficient (at least 3.0) for remote control. By looking at respective video clips of scenarios 3 and 4, it can be seen that both suffered from rainy weather and poor light conditions. In addition, not always the highest quality level was considered as driveable, e. g., as in scenarios 6 and 9. This has to be analyzed further but indicates that the utilized VMAF model was not fully suitable for teleoperated driving. Thus, with this model it was not possible to state that values above a certain VMAF scale are usable for teleoperated driving per se.

In addition, a relation between *Controlling* and the rated video quality (column *Rating*) could be revealed. The MOS was at least 3.0 when participants would trust themselves to remotely control a vehicle. However, this again was not true in all cases. In scenarios 6 and 9 ratings above 3.0 were rated as not sufficient for remotely controlling a vehicle, although both got a MOS rating of at least 3.0 in the specific scenario. This also needs to be analyzed further.

To related the obtained results to real-world bandwidth constraints, we estimated the minimal required bandwidth for each scenario based on the lowest VMAF rating sufficient to to remotely control as vehicle. The mapped results can be seen in Table 3.5 and range from 280 kbps to 832 kbps, considering only values that were generated with at least real-time encoding speed. However, the mapped bandwidth does not include the bandwidths for scenarios 3 and 4, as no values could be specified based on the ratings. They will be somewhere above 3.35 Mbps (scenario 3) and 1.04 Mbps (scenario 4), requiring a greater quality level than presented.

Scene	**Minimal Bitrate**	**Scene**	**Minimal Bitrate**
0	643.81	5	831.92
1	280.00	6	698.29
2	739.58	7	570.82
3	Undef.	8	687.23
4	Undef.	9	299.20

Table 3.5: Comparison of the different minimal VMAF metrics combined with the minimal bitrate by keeping the encoding speed above real-time. Based on the ratings, it was not possible to obtain values for scenario 3 and 4. (Source: [**VI**], ©2020 IEEE)

In general it turned out that the bandwidth requirements are strongly dependent on the environmental condition and the driving situation, which was to be expected. Thus, the estimated bitrates should be treated more as first approximation than as fixed values. They can help to see tendencies and ranges of possible values for the development of future systems, but could be further optimized. Additionally, more types of scenarios need to be evaluated to derive values that can be applied to real-world systems.

Although we showed that, for different types of videos, different levels of compression can be applied that help to reduce required bandwidth, this is only a first step. As we will show in the following Section 3.6.4, more advanced approaches can help to further reduce the bandwidth requirements keeping the scenario driveable.

3.6 Advanced Approach for Bandwidth Reduction (Publication [I])

The basic idea of this approach to further reduce bandwidth consumption consists of splitting a single camera stream into two streams, where important objects and the remainder are separated. While the important part stays untouched, the remainder is further manipulated by applying a bilateral filter to remove details but keep areas and edges (blurring) [143].

This approach was composed of the following three steps that extend each other:

1. In the first step only the lane in front of the vehicle was considered as important.
2. We enhanced this lane-only approach by utilizing two object detection Machine Learning (ML) models (SSD MobileNet v2 320x320 and EfficientDet D7 1536x1536), which were either the fastest or most accurate model available on ModelZoo [96]. Identified important objects were put into the non-blurred stream.
3. Finally, a Field of View (FOV) inspired by 360 degree videos [164] was applied to the stream. This allowed for stronger blurring specific parts of stream, which are outside the FOV.

The blurring itself is twofold and consisted either of keeping the color in the remainder – called Blur-Full (BF) – or of turning the remainder into gray before blurring called Gray Blur-Full (GBF).

3.6.1 Encoding Parameter Results

Based on the ten previously introduced driving scenarios, as shown in Figure 3.20 (Section 3.5.3) in publication [**VI**], our first step consisted of how to further reduce the required bandwidth with H.265 codec parameters while affecting the perceived visual quality at most minimally. Therefore, we altered the encoding parameters *motion estimation search method, motion estimation search range* and the applied *colorspace*. We kept the parameters (resolution, CRF, etc.) that we estimated based on the user study results presented in Section 3.5.3. The first results of the colorspaces *yuv420p* and *gray8* indicated, that the median bandwidth requirements are 353 kbps for *yuv420p* and 380 kbps for *gray8* in BF. Therefore, we used only

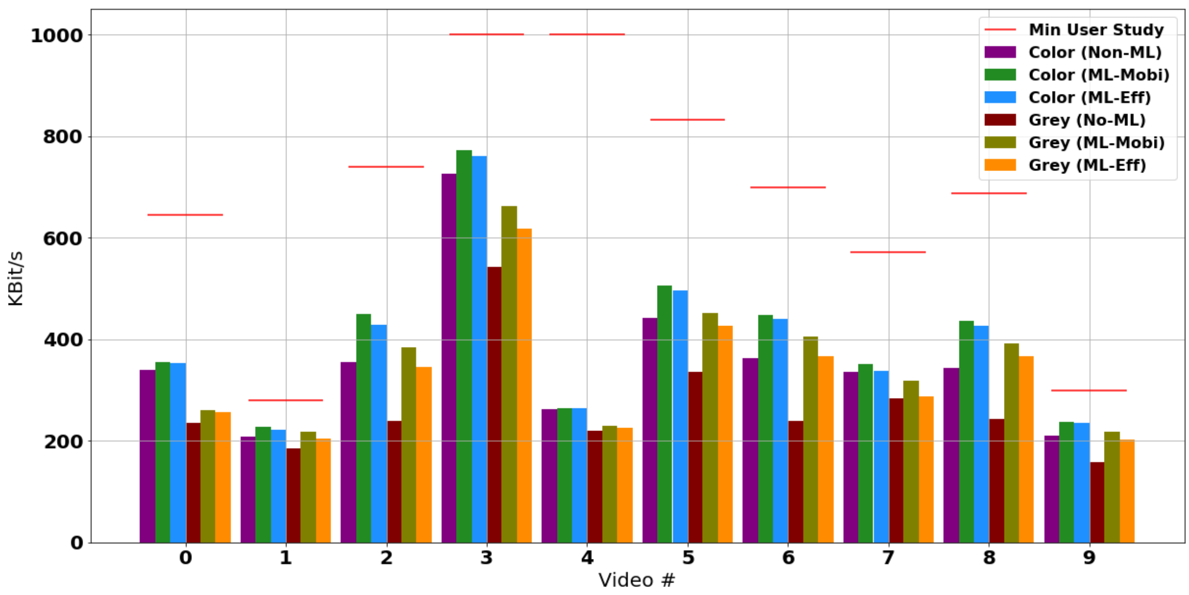


Figure 3.21: Comparison of the lane-only and ML results, with the red bars indicating the the bandwidth-requirements of the previous section (Table 3.5). (Source:[**I**],©2022 IEEE)

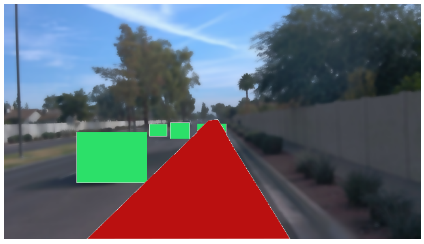
the lower-bandwidth *yuv420p* colorspace. The next parameter that we altered consisted of the motion estimation method, where we compared *hex*, *umh*, *star*, *sea* and *full*, which are ordered based on their execution speed [127]. They differ in the way on how the stream is searched for motion vectors and thus apply different search patterns [127].

In BF median the bandwidth ranged from 340 kbps for *full* to 354 kbps for *hex*. Although the *full* motion-estimation search method provided the best results (340 kbps) within our measurement, it was extremely slow (7 fps for *full* compared to 33 fps for *umh*) and thus not feasible for teleoperated driving. We instead used the second best option *umh* with 352 kbps further on. Finally, the motion estimation search range, indicating the maximum range of the motion search in pixels [37, 127], was altered between the values *0, 8, 16, 32, 57, 64, 128, 256* and *512*. The search range of *57* (H.265 default) led to a median bandwidth of 341 kbps, which was the same as for larger search ranges and thus we used it further.

3.6.2 Stream Manipulation for Bandwidth Reduction

With the encoding parameters being established, we drew our attention to a more advanced approach. As can be seen in Figure 3.22, our main idea was splitting the camera stream into the

(a) Mask



(b) Remainder

(c) Combined

Figure 3.22: Example of the approach to split into mask (a) and remainder (b). The area indicated by red is the one of the mask with the lane-only approach, while green together with red indicates the ML approach area. (c) shows how the stream will be transmitted eventually. (Source: [**I**], ©2022 IEEE)

remainder (Figure 3.22b) and the *mask* (Figure 3.22a). The red marker indicates the important objects (*mask*) for the lane-only approach. We applied the bilateral filter of OpenCV [138] that is "highly effective in noise removal while keeping edges sharp" [138] to the *remainder* and kept the *mask* part as is. After we processed the two individually, both were combined (Figure 3.22c) again.

Our first approach only considered the lane in front of the vehicle as being important. The initial results in Figure 3.21 are shown in purple (BF) and maroon (GBF), with the red horizontal lines representing values obtained in the previous section. Since we could not provide an estimate for scenarios 3 and 4 in Section 3.5.3, they were optimistically assumed to be 1 Mbps. The lane-only approach was always below the red line and thus an average improvement of

Figure 3.23: Difference between the two ML approaches. Blue indicates areas detected by MobilNet, green indicates areas detected by EfficientDet. (Source: [**I**],©2022 IEEE)

317 kbps (BF) and 407 kbps (GBF) could be achieved.

In the next step, we utilized ML-based object detection to detect important objects such as pedestrians or other vehicles that are not in the driving lane in front of the vehicle, but also relevant for the teleoperator's decisions (green rectangles in Figure 3.22).

Therefore, we applied the two ML models *EfficientDet D7 1536x1536* and *SSD MobileNet v2 320x320*. Following [96], the first model has a Common Objects in Context (COCO) mean Average Precision (mAP) of 51.2 at a frame-speed of 325 ms, while the second model has a COCO mAP of 20.2 at a framespeed of 19 ms. This allowed us to compare the fastest and the most accurate model of ModelZoo [96]. An initial example of the difference between the two models is shown in Figure 3.23, where blue and green markers indicate the difference. If setting the detection threshold to 0.45, the average count of detected objects (independent if correct or not) during the 20 s video sequence across all scenarios differed between about 10 detected objects for *MobileNet* and about 6.9 detected objects for *EfficientDet*.

The results of *MobileNet* are shown in Figure 3.21 by green (BF) or olive (GBF) and in-

Figure 3.24: Example for the FOV as used in the proposed approach. (Source: [**I**],©2022 IEEE)

dicate an average improvement of 270 kbps for BF and 321 kbps for GBF by comparing it to the original bandwidth requirements revealed in Section 3.5.3. However, in contrast to the lane-only approach, the ML approach required 46 kbps (BF) and 86 kbps (GBF) more. *EfficientDet* is indicated by blue (BF) and orange (GBF) in Figure 3.21 and came up with an average improvement of 279 kbps (BF) and 345 kbps (GBF) in contrast to the original bandwidth requirements. Again, by comparing it to the lane-only approach, 38 kbps (BF) and 62 kbps (GBF) were required additionally. Through the comparison of the bandwidth requirements of both models, it could be seen that *EfficientDet* in average required 9 kbps (BF) or 24 kbps (GBF) less than *MobileNet*. The additional bandwidth required in the ML approach stems from the fact that fewer areas of the image can be blurred for the ML approach, and thus can potentially be compressed less effectively. The difference between *EfficientDet* and *MobileNet* can be explained by the different number of detected objects.

In as the next step, we applied the FOV approach. The main idea was based on the fact that specific parts of the stream are sharply focused by humans, while everything around them is out of focus. Thus, we stronger blurred outer areas with the bilateral filter of OpenCV as can be seen by example in Figure 3.24. Therefore, we changed the original parameters of the

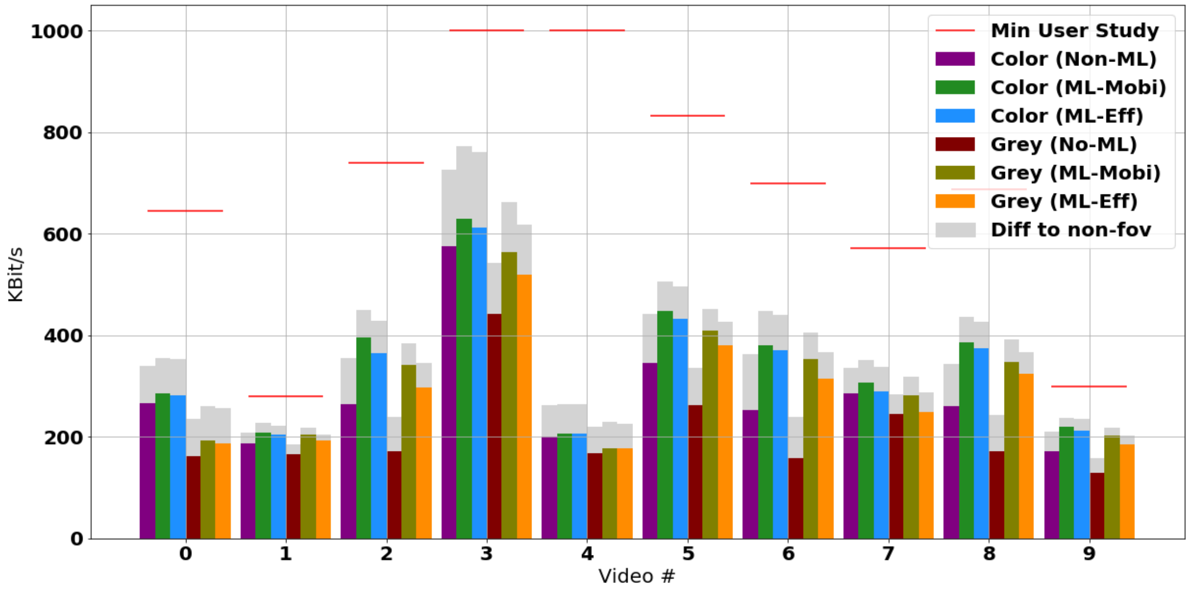


Figure 3.25: Comparison of the field of view lane-only and ML results, where the red bars indicate the bandwidth requirements identified by [**VI**] (Table 3.5). The gray bars show the difference between the non-FOV (gray) and the FOV (color) approach. (Source: [**I**], ©2022 IEEE)

bilateral filter (diameter = 25, sigmaColor = 125 and sigmaSpace = 250) to ones that resulted in a stronger blurring (diameter = 200, sigmaColor = 225 and sigmaSpace = 250). The driving lane and – in the next step – important objects were not be blurred by the FOV approach.

In contrast to the non-FOV lane-only approach, the improvements were in average about 77 kbps for BF and 60 kbps for GBF (see Figure 3.25), showing that the application of the greater blurring can support the encoder in reducing the required bandwidth.

In the next step, we utilized both the *MobileNet* and the *EfficientDet* in the FOV approach. The results of the *MobileNet* showed an average improvement of 329 kbps for BF and 368 kbps for GBF compared to the original bandwidth requirements. The average additional required bandwidth in comparison to the lane-only FOV was about 65 kbps and 100 kbps for BF and GBF, respectively. If conducting a comparison on the same model but without the FOV, the FOV approach provided an average improvement of 58 kbps for BF and 47 kbps for GBF. For *EfficientDet*, an average improvement of 340 kbps for BF and 393 kbps for GBF in comparison to the original bandwidth could be seen. In relation to the equivalent FOV with *MobileNet*, *EfficientDet* required 12 kbps (BF) and 25 kbps (GBF) less. By comparing the model's results to the approach without FOV, an average improvement of 61 kbps (BF) and 48 kbps (GBF) was detected. This again shows that the FOV approach can help to reduce the bandwidth requirements. The ML-based approach also shows that blurring smaller areas of the image increased the required bandwidth eventually. In order to test the usability of the generated video clips, we conducted a user study.

3.6.3 User Study

It was also important to validate the previously calculated results with regard to their applicatbility in real-world. Therefore, we conducted an online survey in which participants had to rate the driveability and the perceived video quality in form of MOS. In general, every participant had to rate $n = 20$ different video clips, that were randomly chosen out of the previously analyzed 192 video clips, which consisted of 24 videos per scenario. Scenarios 3 and 4 were not part of the study, as previous results already indicated that codec only compression led to no driveable rated setting.

At first we had to remove invalid participants by finishing, completion time and used device. Basically we removed all participants that did not finish the study. Additionally, we removed participants who completed the study in a time that was less (including a margin for reading the tasks) than the total time of all videos they were asked to watch and rate. Finally, we also removed all participants that participated with smartphones. Thus, 226 valid participants remained. This led to a number between 16 and 30 ratings per video clip. The first overall results on all video clips showed that the driveability rating was *Rather No* and the MOS with 2.5 was between *Poor* and *Fair*. In more detail, 57 videos (about 30%) out of the 192 were rated at least *Rather Yes* and thus rated as driveable. The 57 combinations could be split into 35 combinations that were ranked driveable for the *basic compression* (highest quality codec settings acting as baseline), while 22 combinations were ranked sufficient for the *study compression* (basic codec settings as obtained from previous work [**VI**], Section 3.5.3).

The *basic compression* did only act as baseline, if a scenario was not rated driveable with the applied *study compression*. Every scenario had at least one combination of parameters $(fov, ml, color)$ rated driveable with the *study compression* and thus the *basic compression*

Area	Weather	Light	FoV	ML	Color	Drive	MOS	Bitrate (kbps)
suburban	sunny	day	fov	eff	col	2.0	2.76	281
suburban	sunny	day	fov	eff	gre	2.0	2.0	186
suburban	sunny	day	fov	mobi	col	2.0	2.87	285
				.				
				.				
urban	rainy	day	nofov	mobi	gre	2.0	2.0	406
urban	rainy	day	nofov	noml	col	2.0	2.29	363
urban	rainy	day	nofov	noml	gre	1.0	1.38	240

Table 3.6: Example of a simple lookup table consisting of the input parameters in gray and the potential resulting combinations in green. (Source: [**I**], ©2022 IEEE)

was not required and thus neglected. Compared to the baseline values from the previous Section 3.5.3, the video clips of the *study compression* that were rated as drivable require 247 kbps less bandwidth. The average driveability rating was *Rather No*, while the MOS was 2.37. However, the combinations rated driveable were different, so it would be difficult for a teleoperated system to choose the most suitable combination. Therefore, we introduced a proposal system that assists the teleoperator in this process.

3.6.4 Adaptive System

As discussed above, the selection of the ideal teleoperator-specific combination of $(fov, ml, color)$ with the respective codec settings can become a complex task. Different environmental conditions will require different settings and thus, our idea built upon these aspects by setting up a lookup table. Such a lookup table can be built as shown in Table 3.6. Based on measurable environmental conditions, the table could contain the ideal approach parameters (e. g., fov, ml, color), the driveability rating, the MOS values and the achieved bitrate. A table applicable in practice could be built based on more complex user studies and then be updated frequently by the experience gathered throguh real-world drivings. Codec parameters are not part of this table, because the table is only meant as an example and therefore does not contain all possible parameters.

Our proposed algorithm consists of 4 steps, requiring available bandwidth, current operator and environmental conditions as input.

Step 1: It is checked whether an advanced approach is required, i. e., if available bandwidth is above minimal compression values. If this is the case, specific codec parameters are returned and the algorithm terminates.

Step 2: Otherwise, the advanced approach comes into play, selecting a combination that matches the given environmental conditions, were rated driveable and require less bandwidth than the available bandwidth. Based on the approach, multiple results are present, e. .g., five different combinations in this approach.

Step 3: If this is not sufficient, i. e., less than five combinations were found, missing entries are filled with combinations that were not rated driveable, but have the highest MOS. This will be marked with a hint to inform the teleoperator.

Step 4: If there are still less than five combinations, combinations with a bitrate above the available value are presented with a hint, being sorted ascending by bitrate.

In order to allow the teleoperator or the system to chose and react properly, the resulting combinations below minimal compression will be sorted in one of the groups: *driveable*, *potentially driveable with speed adjustment* or *above available bandwidth*. Groups are dependent on the output of the algorithm. To apply this adaptive system in real-world applications, the network has to be measured frequently and respective transitions between the combinations need to be smoothly, i. e., the teleoperator should only notice it marginally as otherwise he could become distracted.

In general, we have shown that bandwidth could be decreased by splitting up the video in two parts and applying different filters on it. We confirmed the results with a user study and revealed that for every scenario at least one combination was rated as being sufficient for teleoperated driving. Due to the huge number of potential combinations of parameters, we proposed an adaptive system that can support in selecting always the most suitable combination of parameters. Although the results of this work are promising, we only used a limited set of video clips. In order to transfer this system into a real-world application, further analysis with more video clips and more participants is required.

3.7 Summary

The results of the real-world measurements show that teleoperated driving is feasible with contemporary networks most of the time, but outliers exist. In the majority latency was below 250 ms. The same applies to uplink and downlink, where values were above 3 Mbps for uplink and 0.25 Mbps for downlink. It also turned out that handover can have a negative impact on the network performance by increasing latency and decreasing throughput. This also applies to signal strength, i. e., the better the signal strength, the higher the throughput. For the distance between teleoperation station and remote vehicle it can be seen that larger distances lead to an increased fluctuation in latency.

The introduced driver support system offers an approach fore dealing with different levels of latency, bandwidth and packet loss in an efficient way by addressing the distance to the vehicle ahead, the stopping distance and driving through curves. We showed that up to a certain level, bandwidth and latency can be reduced to latency-related issues and latency can be handled efficiently by adjusting the remote vehicle's speed. We revealed in a calculation that for real-world network performance the speed adjustment is not drastic (3 km/h) slower as with an usual non-remote driving for an inner-city route. The remote operated vehicle thus would not pose a traffic safety problem.

To evaluate the performance of different conditions, we developed OpenROUTS3D, a driving simulator for the needs of teleoperated driving, offering the ability to extend it easily. We used it for testing different approaches and claimed its usability by being used as the main system for the user study regarding latency, where it did not show any flaws. Objective and subjective measurements of the latency-based user study indicated that latency should be kept as low as possible but at least below 300 ms to allow for safe remote control. This prevents the teleoperator from being put under too much pressure or stress. It also makes sense to apply a system that tries to keep latency as constant as possible, since varying latency increases the perceived workload.

When evaluating the perceived video quality of different driving situations with a user study, we found that the results vary greatly depending on the driving scenario and weather conditions. Results also showed that the VMAF model can be used as basic indicator whether a video quality is sufficient or not, although it is not ideally fitted. However, some outliers exists, e. g., the influence of bad weather and bad light conditions. Mapping the obtained results on minimal bitrates, the scenario dependence reveals again. The overall range consisted of values from 280 kbps to 832 kbps. However, there are also scenarios where we could give no real estimate, as no presented quality level was rated sufficiently. We further showed that bandwidth can be reduced additionally by utilizing a more advanced approach. This approach splits the stream into important and less important objects. We did this by applying a bilateral filter on the less important ones. Results showed an average reduction of the required bandwidth of up to 467 kbps, which corresponds to about 34% of the original bandwidth. Based on the driveability rating of the survey, the improvement across all driveable rated video clips was settled at 247 kbps (60% of the originally required bandwidth), where at least one combination of parameters per scenario was rated sufficient for teleoperated driving.

Altogether, the results are promising for teleoperated driving and guide the direction that it could be used in *everyday's traffic scenario* in future. Our measurements revealed that contemporary cellular networks are sufficient most of the time. We showed that the basic approaches of whitelisting road sections and adjusting the remote vehicle's speed could contribute to a safe remote drive. The results of the user study suggest that latency can have a major impact on objective and subjective measurements and should therefore be at least less than 300 ms in real-world applications. Finally, we revealed that bandwidth can be reduced efficiently by applying basic encoder settings and a more advanced approach differentiating between important and less important objects.

4 Discussion and Future Work

With this thesis we addressed the obstacles induced by cellular networks, especially the influence of bandwidth, latency and packet loss in a way that can help to allow the use of teleoperated driving in real-world traffic scenarios. Based on the five main underlying research questions **RQ1–RQ5**, we took the goal of making teleoperated driving applicable in *everyday's traffic scenarios* one step further with this thesis.

RQ1: Is teleoperated driving feasible with contemporary cellular networks?

One crucial step on allowing remote operations on real-world traffic scenarios is to measure whether it is possible at all to remotely control a vehicle with existing cellular networks, i. e., if measurements show if the network can meet the specific requirements. To this end we measured the German cellular network of one provider. We found that with a median of about 55 ms in the Round Trip Time (RTT), the minimal required latency of 250 ms could be undercut in the majority of the measurements (96%). This holds for the measured uplink and downlink speed. Downlink was, depending on the measurement setup, in about 95%–99% of the measurements above the minimal required bandwidth of 0.25 Mbps. Uplink was, depending on the measurement setup, sufficient for teleoperated driving in about 87%–98% of the measurements. We witnessed that a handover between two cells can have a negative impact on latency and bandwidth. It also turned out that the distance between teleoperation station and remote vehicle can increase fluctuation in latency. We also showed that a basic approach such as whitelisting road segments could be a support for using teleoperated driving. Although the results are promising, the work is limited by its amount of measurements and the geographical area that was covered. Measurements in different areas and/or with other providers at different times of the day could lead to divergent results and, of course, the infrastructure improves both in performance and coverage. This will also lead to new results.

RQ2: Is it possible to achieve a safe remote control of a vehicle on typical roads despite latency using basic algorithms?

One major aspect in teleoperated driving is the need to ensure that the remote vehicle can always be controlled safely. This means that the teleoperator is always in full control of the remote vehicle, i. e., remote control is ideally not an additional risk compared to a non-remote operation. Our approach of addressing this topic follows the idea, that one can try to counteract actual latency by adjusting the remote vehicle's speed. By adjusting the allowed maximum speed in teleoperated driving based on current network parameters, it could be possible to allow for a safe remote drive. It is theoretically conceivable to always keep enough distance to vehicles ahead, keep the stopping distances as without latency and also drive safely through curves. Based on real-world latency measurements, the theoretical feasibility of the approach was confirmed. We have shown that the approach would not leave remotely controlled vehicles as obstacles in traffic by being too slow. When considering the measurements of the city-center of Ingolstadt, the speed decreasing on the specific route would only be about 3 km/h. Finally, we introduced basic route planning, allowing teleoperated driving only in areas that provide sufficient network performance, i. e., being measured or pre-computed based on specific algorithms or Artificial Intelligence (AI). Although the approach was only shown in theory, it could be adjusted to fit the needs of a real-world system. Nevertheless, this is only one first step into this direction as the investigation lacks a real-world test of the proposed speed reduction. This in itself requires a more accurate vehicle model that can approximate the actual vehicle's performance as close as possible.

RQ3: What is the influence of variable and fixed latency on teleoperated driving performance and subjective assessment?

In order to validate algorithms in teleoperated driving, driving simulators can be a useful. Since the detailed search for a suitable driving simulator did not lead to a satisfactory result, we developed an Open Source driving simulator called Open Realtime OSM- and Unity-based Traffic Simulator 3D (OpenROUTS3D)

In order to address latency induced issues in teleoperated driving, we wanted to investigate how latency influences the driving performance and the subjective perception of potential teleoperators. We also wanted to see if we can identify first potential latency bounds for specific driving situations. We conducted a user study utilizing OpenROUTS3D, in which participants had to drive different scenarios with different levels of fixed or varying latency to investigate the impact of latency. The results of the user study showed a clear tendency that higher latencies make remote driving more challenging, confirming the expectations. However, the impact of latency is highly scenario-dependent. It has lower effects in scenarios that are driven slowly, e. g., a parking scenario, while other scenarios which are driven faster show greater impacts, e. g., remote driving on a rural road. Measured objective and subjective values indicated that the experienced workload increases and the driving performance decreases with increasing latency. In addition, no significant difference between constant and varying latency could be seen within our setup, in which the road configurations where different between both driving situations. Overall, it turned out that a RTT of about 300 ms, including latency induced by the network and sensors with actuators, could be an indicator to mark the upper bound for teleoperated driving without an assistance system. Although the results of our user study can guide a specific direction, they are only an indicator. The limited number of 28 participants and a limited set of driven scenarios only allow to identify a first trend. In addition, participants drove in a safe environment, in which they could not destroy anything or harm anyone. The study was also focused only on latency, i. e., the impact of bandwidth and packet loss was not considered, but we would expect this to also influence the driving performance.

RQ4: How far can video streams be compressed to still allow for safe teleoperated driving?

Another part that we addressed was the efficient reduction of data rate. We applied state of the art compression techniques to reduce the required uplink for the video stream of the remote vehicle's environment. However, the level of compression cannot be arbitrary high as a minimal visual quality must be kept during remote operation in order to allow the teleoperator to sense the environment and identify important objects. Therefore, we modified multiple compression parameters such as Constant Rate Factor (CRF), resolution, bitrate, preset/tune, and codec and applied them to ten different driving scenarios in order to generate multiple combinations of parameters per scenario. Using an automated clustering based on Video Multi-Method Assessment Fusion (VMAF) scores, five videos per driving scenario were presented to participants in an online-survey, in which they had to rank the driveability and the perceived quality by means of the Mean Opinion Score (MOS). It turned out that the minimum required quality strongly depends on the environmental conditions, especially weather and lighting. This is also reflected in the minimal required bitrates, which ranged between 280 kbps and 832 kbps within our selected videos. With the limited number of 70 valid participants and a limited set of scenarios, the results can be seen as first indicators to guide the way. In addition, the videos were presented at 20 Hz, but were recorded at 10 Hz, i. e., the scenarios look faster than they actually are. Finally, participants used their own setups with different monitors, which in general can have a influence on the perceived quality.

RQ5: How can the required bandwidth for uplink video streams be decreased?

Finally, we investigated whether more advanced approaches could help to further reduce the uplink requirements. The approach consisted of distinguishing between objects that are more or less relevant to driving. Relevant objects are, for example, the driving lane in front of the vehicle and all objects that might be of interest due to their movement, e. g., other vehicles, pedestrians, etc. The presented approach identifies the lane in front of the vehicle and important objects, e. g., using machine learning, and splits the single stream into a mask for important objects and a remainder for everything else. This allows keeping all parts of the mask within a level of compression that was already rated sufficiently, while everything around could be compressed stronger, i. e., by applying a bilateral filter to keep edges but remove small details. This helped to further reduce the required bandwidth and as such to allow teleoperated driving in a larger area. For validation, we conducted an online survey in which participants had to rate the perceived quality (MOS) and the driveability. Results showed that an improvement of about 247 kbps (60% of the originally required bandwidth) across the investigated scenarios could be achieved. Based on those findings, we developed a theoretical system. This system helps based on the specific teleoperator to select the most suitable parameters. Although the approach shows promising results, the work should be seen as another step within the field of teleoperated driving. One major drawback is the limited number of videos and the combinations of parameters that were tested. Additionally, the limited number of participants and the use of their own devices can have a strong influence on the perceived quality and the respective ratings.

With various important topics being addressed by this work, it is one further contribution towards teleoperated driving in real world scenarios. In the process, the work shown here has already been taken up by other research groups to identify a research gap or a address a specific topic of teleoperated driving and therefore proofed its relevance within the field. As an example, measurements with respect to remote operation were also conducted e. g., by Burke et al. [24] and Gaber et al. [59], either confirming the results or considering our findings as base for their research. The idea of reducing image quality by video stream encoding for teleoperated driving was considered by the work of Hofbauer et al. [84], where they applied a camera view prioritization. The priority indicates, based on the available bandwidth, how strong each camera stream needs to be compressed.

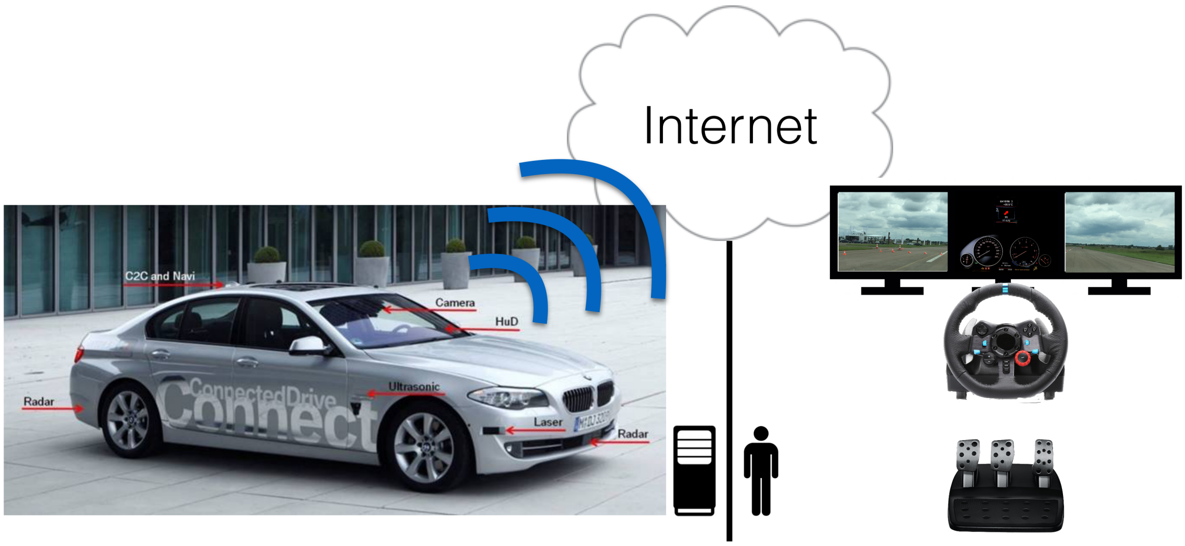


Figure 4.1: Teleoperated system, with the vehicle on the left side and the teleoperation station on the right side. (Based on [165], Source: [**II**], Reproduced with permission from Springer Nature)

Recapping Figure 1.5 here as Figure 4.1, it can be seen that the focus of this thesis was put in the middle part of the shown setup, the cellular network. This mainly consisted of addressing bandwidth and (network-)latency-related issues and provide first steps toward real-world application of teleoperated driving.

However, there are more areas where research is required. When considering the remote vehicle – left part of Figure 4.1 – various interesting topics are to be covered. Those topics mainly consist of required actuators, sensors and pieces of software that need to installed on remote vehicles to allow for a safe remote operation. With focus on the right part of Figure 4.1, the teleoperation station, research with focus on Human-Machine Interface (HMI) is required. This mainly consists of ways for displaying relevant information such as the remote vehicle's environment and network-related information to the remote operator and approaches on how enable an proper way of controlling remote vehicles. The security characteristic also play an important role in the trustability of remote control. Overall, other aspects such as legal issues, liability questions and monitarization need to be considered, i. e., the question on how to earn money with teleoperated driving needs to be answered. If these issues are not addressed properly, teleoperated driving may not be deployed despite technical feasibility. This also counts for considering the needs and potential safety concerns of passengers and gain their acceptance. They need to feel comfortable being driven remotely or having remotely driven cars as traffic participants next to them. Without this acceptance no one will ever use remote driving.

As can be seen, many topics are still open an need to be resolved in order to eventually allow teleoperated driving in *everyday's traffic scenarios*.

4.1 Limitations

For real-world results as presented in Chapter 3.1.2, the main drawback consists of the limited amount of measurements. By conducting them in different areas, the results may become better or worse, depending on the utilized provider and the available coverage. Even in the measured areas deviations in the network performance will occur as changes in the infrastructure and the network workload are likely to happen. Therefore, the utilized networks had to be treated as black box within this paper, i. e., it was not possible to see whether the network or the cell was busy or not. Nevertheless, the amount of measurements and the resulting findings could be used to answer the addressed research questions and to allow to get a first assessment of the network performance with respect to teleoperated driving. In order to take this into real-world applications, more detailed measurements are required. This will help to better understand the deviation of the network parameters and to contribute to the proposed whitelisting approach.

Although the assistance system for teleoperated driving as shown in Chapter 3.2.5 helps to support with different levels of latency, its main drawback is the missing validation in a real-world remotely controlled vehicle with usual traffic. In order to build such a system, the stationary treatment of braking and accelerating by respective values needs to be more realistic, i. e., the vehicle will not suddenly change speed but slowly increase or decrease it in real-world driving situations. This can be solved by using more accurate models, that are specific to individual types of vehicles. We also kept the impact of the vehicle dynamics simple, i. e., the influence of parameters such as wind or different types of vehicles was not considered, which again requires more detailed vehicle-specific models. Additionally, the approach can only work within specific bounds, i. e., video compression cannot reduce the bandwidth arbitrarily without dropping visual quality below an unsuited level. Nevertheless, we have shown that the assistance system mainly consisting of speed adjustments can work and if real-world vehicles are allowed to be operated remotely, it is easy to adjust the specific parameters and apply the proposed solution.

When conducting simulator-based user studies, as for the teleoperated driving-related latency in Chapter 3.4.4, the number of participants and driven scenarios is crucial. With a small number of 28 participants and a limited number of driven scenarios and latencies, the results can be seen as a trend. However, it is hard to provide a specific conclusion, which will be valid for all potential real-world driving situations. This is potentially supported by the fact, that participants will drive different in virtual environments, as they subliminally know that they cannot be harmed and deploy no harm. In addition, skilled and specially trained drivers will likely achieve different results and even various setups such as using virtual-reality systems might lead to varying outcomes [85]. Finally, we tested latency in an environment without other traffic participants, i. e., for real-world scenarios the combination of latency together with bandwidth and packet loss in scenarios with other traffic participants might lead to different results. However, the results indicated that latency should be kept below 300 ms to allow for a safe drive. In order to further validate those results, more detailed user studies are required. Those studies will need to investigate more complex scenarios, e. g., with other traffic

participants. It would also be important to consider a special training for remote operators, to see if this can help to better react on latency and how this training should look like.
The same limitations regarding the limited number of 70 valid participants and the small set of scenarios also exist for the basic video compression presented in Chapter 3.5.3. Another limitation was the presentation of the video clips in 20 Hz, being recorded at 10 Hz, which made scenarios look faster than they actually are. In addition, the non closed-loop character of the study with different setups/monitors may lead to slightly different results than conducting the user study in close-loop with the same system. A great number of ratings per video can counteract this effect to a certain extend and unsuitable resolutions were already filtered initially. Nevertheless, the results of the study are valid and the basic findings that weather and lightning conditions have a strong influence on the ratings are interesting, i. e., the bandwidth calculations can be used to indicate potential ranges of bandwidth requirements for real-world systems. The main drawbacks of this study can be solved by performing such video-ratings in a controlled environment, e. g., a laboratory. Additionally, with an increased number of participants and rated driving scenarios, the results will become more stable and can be used easier for transfer to real-world systems.
Finally, a major limitation in the work of further reducing the required bandwidth by blurring as presented in Chapter 3.6.4 is again the limited number of video clips and combinations that we tested, i. e., with different parameters a greater improvement of bandwidth might be achieved. The preprocessing also adds some cost on the latency, e. g. 0.33 s of additional time for each frame with EfficientDet. Possibly not all preprocessing steps can be conducted because of an already comparably high overall latency. In addition, we validated the presented results with a user study, but with a limited number of participants. Every participant rated the video clips on their own devices, which can lead to different ratings due to variations in the display quality. However, this can be counteracted with the great number of ratings per video. Nevertheless, results can be used as baseline for developing approaches suitable for real-world use-cases, which then can help to validate and improve the approach.

Although there is a number of limitations, most of them can be handled by more measurements, real-world test drives or larger user studies and they do not have a strong influence on the answer to the main question raised by this thesis. All work leads to the same direction,

i. e., the results can be used to proof the claim that teleoperated driving can be possible *in everyday's traffic scenario*, at least based on the results we presented in this thesis. When deploying it to real-world vehicles, some adjustment and fine-tuning will be required, but overall the proposed approaches can be utilized.

4.2 Directions for Future Work

Based on the results presented in this thesis, future work can address and extend several topics. The major parts derived from the topics of this thesis consists of putting the developed approaches into real-world applications and conduct additional user studies. This means that the approaches need to be adjusted based on specific vehicles and thus may reveal that one change or another is further required to allow the utilization *in everyday's traffic scenarios*.

With focus on the greater overall system of teleoperated driving, future work will need to address pieces such as hardware and software installed on the remote vehicle. This mainly consists of actuators and sensor that are required for operating a remote vehicle, e. g., camera, Light Detection and Ranging (LiDAR), but also needs to consider hardware that is needed to manage connectivity, e. g., modems. All of these parts need to be reliably connected and supported by specific software. Selecting suitable pieces of Hardware and reliably integrating them into a vehicle is one of the portions that need to be investigated. By considering the development of future vehicles in huge automotive companies, the consideration of teleoperated driving could be a strong benefit. This can help to directly integrate required hardware and interfaces into the vehicle and select appropriate sensors during the development. Addressing sensors, examining what data the teleoperator needs at what stage of the remote drive and how existing sensor data can be fused to be supportive needs to be part of research. For example, in parking situations, ultrasonic sensors and a parking camera with a special angle may be of interest, while the additional application of LiDAR could be useful when driving within a city. In addition, other sensors like accelerometers or data from the Electronic Stability Program (ESP) will be required to gather all the information that is required for achieving a certain level of telepresence. This is also associated with the place of the sensor data fusion. Since this can be done on the vehicle but also on teleoperation station, it needs to be investigated what is more beneficial. By performing the sensor data fusion in the vehicle, the required bandwidth may be decreased, but the teleoperator is probably not able to access specific sensor data directly. In contrast, when all data is transmitted to the teleoperation station, the bandwidth requirements will increase.

It is also important to further investigate the ideal setup for a teleoperator, e. g., which combination of displays, feedback and controllers really supports the human and allows him to perform a safe remote drive. One of the areas that need to be covered is how to best achieve telepresence. In combination with the aforementioned different set of sensors, the question arises as to how depth perception can be achieved. One direction could be the combination of LiDAR data and stereo-cameras to allow a accurate 3D projection of the environment with Head-Mounted Display (HMD) or a multi-monitor setup. In addition to the visual component of the perception, haptic and motion awareness are important. Haptic feedback to the teleoperator could be provided by force-applying steering wheels (e. g., Logitech G29 [117]) and thus allow the teleoperator to react to wheel-slip indicated by the ESP. One approach to allow the sensing of the remote vehicle's actual motion could be based on motion platforms (e. g., motion platform v3 [134]).

Future research for teleoperated driving needs to further cover the topic of networks. More network measurements, especially considering the performance of the 5G network, being currently deployed in Germany [56], would be interesting [27]. With the claims of 5G some of the proposed approaches might be used differently, e. g., lower latency will allow driving in greater areas, higher bandwidth will require less compression, etc. Nevertheless, it would also be interesting to see whether 5G will change they way teleoperated driving could be used, or if the real-world benefits are not as great compared to LTE-Advanced. In addition, it would be interesting to investigate how huge the benefit of utilizing multiple service provider in parallel is and if the parallel use of different technologies (e. g., 5G and Long Term Evolution (LTE)) could provide support in certain driving situations. This could be covered by performing further detailed real-world measurements and test drives with remote vehicles. One could also think about applying IEEE 802.11 Wireless Local Area Network (WLAN) or installing their own 5G campus network [147] instead of using public cellular networks in certain situations. For example, a dedicated network could be applied in parking lots allowing a more controlled environment.

The future use for teleoperated driving could be strongly driven by AI-based approaches to support the teleoperator in all driving situations. This could mean that driver assistance systems on the remote vehicle remain activated for as long as possible. Additionally, it needs to be investigated whether this also can be applied at the teleoperation station. By adjusting or filtering steering commands that could be dangerous to the remotely controlled vehicle, directly at the teleoperation station, teleoperated driving could benefit. Nevertheless, this needs to be researched and tested thoroughly.

Besides those technical aspects, further directions for future work arise from legal, liability, trust and monitarization topics. Therefore, research could focus on the trustablity of teleoperated driving and related topics, e. g., how to gain the trust of potential customers in such a system. Focus also should be put on how to monitarize remote operations, answering the question who is about to pay how much for such a system. One could think of different types such as pay-per-use, flatrates, etc. Legal and liability issues also need to be addressed, as a real-world application is only possible with specific laws and defined rules on liability.

Although a lot of research and development effort still needs to be put into teleoperated driving. It can still be seen as a supporting and extending piece in the development of autonomous systems. With the results presented in this thesis, we showed that the application in real-world scenarios could be feasible. Considering the multitude of potential uses and results we presented in this thesis, it can be assumed that teleoperated driving has an interesting future ahead and will be beneficial in many areas.

Publications as First Author

[I] Stefan Neumeier, Vaibhav Bajpai, Marion Neumeier, Christian Facchi, and Jörg Ott. "Data Rate Reduction for Video Streams in Teleoperated Driving". In: *IEEE Transactions on Intelligent Transportation Systems* 23.10 (Oct. 2022), pp. 19145–19160. ISSN: 1524-9050. DOI: 10.1109/TITS.2022.3171718.

[II] Stefan Neumeier, Christopher Corbett, and Christian Facchi. "A Secure and Privacy Preserving System Design for Teleoperated Driving". In: *Advances in Information and Communication*. Vancouver, Canada: Springer International Publishing, Apr. 2021. ISBN: 978-3-030-12388-8.

[III] Stefan Neumeier and Christian Facchi. "Towards a Driver Support System for Teleoperated Driving". In: *22nd Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct. 2019.

[IV] Stefan Neumeier, Nicolas Gay, Clemens Dannheim, and Christian Facchi. "On the Way to Autonomous Vehicles - Teleoperated Driving". In: *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*. Dortmund, Germany: VDE, Mar. 2018, pp. 1–6.

[V] Stefan Neumeier, Michael Höpp, and Christian Facchi. "Yet Another Driving Simulator OpenROUTS3D: The Driving Simulator for Teleoperated Driving". In: *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. Graz, Austria: IEEE, Nov. 2019.

[VI] Stefan Neumeier, Simon Stapf, and Christian Facchi. "The Visual Quality of Teleoperated Driving Scenarios - How good is good enough?" In: *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. Montreal, Canada: IEEE, Oct. 2020.

[**VII**] Stefan Neumeier, Ermias Walelgne, Vaibhav Bajpai, Joerg Ott, and Christian Facchi. "Measuring the Feasibility of Teleoperated Driving in Mobile Networks". In: *2019 Network Traffic Measurement and Analysis Conference (TMA)*. Paris, France: IEEE, June 2019, pp. 113–120.

[**VIII**] Stefan Neumeier, Philipp Wintersberger, Anna-Katharina Frison, Armin Becher, Christian Facchi, and Andreas Riener. "Teleoperation: The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters". In: *Proceedings of the 11th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '19. Utrecht, Netherlands: ACM, Sept. 2019. DOI: 10.1145/3342197.3344534. URL: https://doi.org/10.1145/3342197.3344534.

Publications as Co-Author

[**IX**] Niklas Hehenkamp, Christian Facchi, and Stefan Neumeier. “How to Achieve Traffic Safety with LTE and Edge Computing”. In: *Advances in Information and Communication*. Ed. by Kohei Arai and Rahul Bhatia. San Francisco, USA: Springer International Publishing, Mar. 2019, pp. 164–176. ISBN: 978-3-030-12388-8.

[**X**] Silas C. Lobo, Stefan Neumeier, Evelio M. G. Fernandez, and Christian Facchi. “InTAS - The Ingolstadt Traffic Scenario for SUMO”. In: *SUMO User Conference 2020*. EPiC Series in Computing. EasyChair, 2020.

[**XI**] Ildiko Tarkanyi, Nicolas Gay, Christian Icking, Lihong Ma, Stefan Neumeier, Graham Hili, and Clemens Dannheim. “Roadside Sensor Optimization in Infrastructure-based Teleoperation of Vehicles”. In: *AmE 2019 - Automotive meets Electronics; 10th GMM-Symposium*. Dortmund, Germany: VDE, 2019, pp. 1–6.

[**XII**] Ermias Andargie Walelgne, Setälä Kim, Vaibhav Bajpai, Stefan Neumeier, Jukka Manner, and Jörg Ott. “Factors Affecting Performance of Web Flows in Cellular Networks”. In: *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. Zurich, Switzerland: IEEE, May 2018, pp. 73–81.

References

[13] ERTRAC Working Group ”Connectivity and Automated Driving”. *Automated Driving Roadmap*. Research rep. ERTRAC, July 2017. URL: http://www.ertrac.org/uploads/documentsearch/id38/ERTRAC%5C_Automated-Driving-2015.pdf.

[14] Bettina Abendroth and Ralph Bruder. “Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort”. In: *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Ed. by Hermann Winner, Stephan Hakuli, Felix Lotz, and Christina Singer. Wiesbaden: Springer Fachmedien Wiesbaden, 2015. Chap. Die Leistungsfähigkeit des Menschen für die Fahrzeugführung, pp. 4–14. ISBN: 978-3-658-05734-3. DOI: 10.1007/978-3-658-05734-3_1. URL: http://dx.doi.org/10.1007/978-3-658-05734-3_1.

[15] IPG Automotive. *CarMaker — IPG Automotive*. https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/. Accessed on 10.04.2019.

[16] AVL. *AVL CRUISE*. https://www.avl.com/cruise. Accessed on 10.04.2019.

[17] F.A. Azis, M.S.M. Aras, M.Z.A. Rashid, M.N. Othman, and S.S. Abdullah. “Problem Identification for Underwater Remotely Operated Vehicle (ROV): A Case Study”. In: *Procedia Engineering* 41 (2012). International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), pp. 554–560. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2012.07.211. URL: http://www.sciencedirect.com/science/article/pii/S1877705812026112.

[18] Vaibhav Bajpai and Jürgen Schönwälder. "A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts". In: *IEEE Communications Surveys and Tutorials* 17.3 (2015), pp. 1313–1341. DOI: 10.1109/COMST.2015.2418435. URL: https://doi.org/10.1109/COMST.2015.2418435.

[19] Daniel Beisel, René Hosse, and Eckehard Schnieder. "Modellbasierte Analyse der Auslegung von Assistenzsystemen". In: *ATZ - Automobiltechnische Zeitschrift* 115.1 (Jan. 2013), pp. 76–82. ISSN: 2192-8800. DOI: 10.1007/s35148-013-0018-2. URL: https://doi.org/10.1007/s35148-013-0018-2.

[20] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner. "Three Decades of Driver Assistance Systems: Review and Future Perspectives". In: *IEEE Intelligent Transportation Systems Magazine* 6.4 (Oct. 2014), pp. 6–22. ISSN: 1939-1390. DOI: 10.1109/MITS.2014.2336271.

[21] Bitkom. *In welchen Situationen wären Sie grundsätzlich bereit, die Kontrolle über Ihr Fahrzeug abzugeben?* https://de.statista.com/statistik/daten/studie/463125/umfrage/geeignete-situationen-fuer-autonomes-fahren/. Accessed on 12.09.2017. Feb. 2017.

[22] Brennanphillips. *ROV Hercules 2005*. https://commons.wikimedia.org/wiki/File:ROV_Hercules_2005.JPG. Accessed on 28.12.2020. July 2006.

[23] Bundesministerium für Verkehr und digitale Infrastruktur. *Entwurf eines Gesetzeszur Änderung des Straßenverkehrsgesetzes und des Pflichtversicherungsgesetzes – Gesetz zum autonomen Fahren*. Tech. rep. Accessed on 03.04.2021. Invalidenstraße 44, 10115 Berlin: Bundesministerium für Verkehr und digitale Infrastruktur, Feb. 2021.

[24] P. J. Burke. "A 4G-Connected Micro-Rover With Infinite Range". In: *IEEE Journal on Miniaturization for Air and Space Systems* 1.3 (2020), pp. 154–162. DOI: 10.1109/JMASS.2020.3018660.

[25] Ryad Chellali and Khelifa Baizid. "What Maps and What Displays for Remote Situation Awareness and ROV Localization?" In: *Human Interface and the Management of Information. Interacting with Information*. Ed. by Gavriel Salvendy and Michael

J. Smith. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 364–372. ISBN: 978-3-642-21669-5.

[26] J. Y. C. Chen, E. C. Haas, and M. J. Barnes. "Human Performance Issues and User Interface Design for Teleoperated Robots". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (Nov. 2007), pp. 1231–1245. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905819.

[27] Xiang Cheng, Ziwei Huang, and Shanzhi Chen. "Vehicular communication channel measurement, modelling, and application for beyond 5G and 6G". In: *IET Communications* 14.19 (2020), pp. 3303–3311. DOI: https://doi.org/10.1049/iet-com.2020.0531. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-com.2020.0531.

[28] Frederic Chucholowski, Tito Tang, and Markus Lienkamp. "Teleoperated Driving Robust and Secure Data Connections". In: *ATZelektronik worldwide* 9.1 (Feb. 2014), pp. 42–45. ISSN: 2192-9092. DOI: 10.1365/s38314-014-0226-x. URL: https://doi.org/10.1365/s38314-014-0226-x.

[29] Frederic Emanuel Chucholowski. "Evaluation of Display Methods for Teleoperation of Road Vehicles". In: *Journal of Unmanned System Technology* 3.3 (2016), pp. 80–85.

[30] Craig Cole. *2021 Mercedes-Benz S-Class is ready for Level 3 automated driving*. https://www.cnet.com/roadshow/news/2021-mercedes-benz-s-class-level-3-autonomous-driving/. Accessed on 27.12.2020. Sept. 2020.

[31] Justin Couch. *Randomation Vehicle Physics*. https://github.com/JustInvoke/Randomation-Vehicle-Physics. Accessed on 20.03.2019.

[32] P. M. d'Orey, A. Hosseini, J. Azevedo, F. Diermeyer, M. Ferreira, and M. Lienkamp. "Hail-a-Drone: Enabling teleoperated taxi fleets". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. June 2016, pp. 774–781. DOI: 10.1109/IVS.2016.7535475.

[33] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G, LTE-advanced Pro and the Road to 5G*. Third. Academic Press, 2016. ISBN: 978-0-12-804575-6.

[34] Konstantinos Dalamagkidis. “Classification of UAVs”. In: *Handbook of Unmanned Aerial Vehicles*. Ed. by Kimon P. Valavanis and George J. Vachtsevanos. Dordrecht: Springer Netherlands, 2015, pp. 83–91. ISBN: 978-90-481-9707-1. DOI: 10.1007/978-90-481-9707-1_94. URL: https://doi.org/10.1007/978-90-481-9707-1_94.

[35] Konstantinos Dalamagkidis. “Definitions and Terminology”. In: *Handbook of Unmanned Aerial Vehicles*. Ed. by Kimon P. Valavanis and George J. Vachtsevanos. Dordrecht: Springer Netherlands, 2015, pp. 43–55. ISBN: 978-90-481-9707-1. DOI: 10.1007/978-90-481-9707-1_92. URL: https://doi.org/10.1007/978-90-481-9707-1_92.

[36] Konstantinos Dalamagkidis, Kimon P Valavanis, and Les A Piegl. *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations*. Vol. 54. springer science & Business Media, 2011.

[37] Dark_Shikari. *x264 FFmpeg Options Guide - Linux Encoding*. https://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping. Accessed on 24.07.2021.

[38] Alex Davies. *Nissan’s Path to Self-Driving Cars? Humans in Call Centers*. https://www.wired.com/2017/01/nissans-self-driving-teleoperation. Accessed on 21.10.2018. May 2017.

[39] Alex Davies. *The War to Remotely Control Self-Driving Cars Heats Up*. https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars. Accessed on 04.04.2019. Mar. 2019.

[40] J. Davis, C. Smyth, and K. McDowell. “The Effects of Time Lag on Driving Performance and a Possible Mitigation”. In: *IEEE Transactions on Robotics* 26.3 (June 2010), pp. 590–593. ISSN: 1552-3098. DOI: 10.1109/TRO.2010.2046695.

[41] Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR). *Unbemanntes Luftfahrzeug*. https://www.dlr.de/content/de/bilder/2012/4/unbemanntes-luftfahrzeug_8467.html. Accessed on 28.12.2020. Dec. 2012.

[42] Sichere Intelligente Mobilität Testfeld Deutschland. *Deliverable D5.5 – Teil A TP5-Abschlussbericht – Teil A*. http://www.simtd.de/index.dhtml/object.media/deDE/8154/CS/-/backup_publications/Projektergebnisse/simTD-TP5-Abschlussbericht_Teil_A_Manteldokument_V10.pdf. Accessed on 05.12.2016. June 2013.

[43] Matthias Dick, Oliver Wellnitz, and Lars Wolf. "Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games". In: *Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support for Games*. NetGames '05. Hawthorne, NY: ACM, 2005, pp. 1–7. ISBN: 1-59593-156-2. DOI: 10.1145/1103599.1103624. URL: http://doi.acm.org/10.1145/1103599.1103624.

[44] K. Doki, K. Suzuki, A. Torii, S. Mototani, Y. Funabora, and S. Doki. "AR video presentation using 3D LiDAR information for operator support in mobile robot teleoperation". In: *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. Herl'any, Slovakia: IEEE, Jan. 2021, pp. 000059–000064. DOI: 10.1109/SAMI50585.2021.9378687.

[45] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[46] Designated Driver. *Designated Driver - Seamless teleoperation. For every situation. For any vehicle.* https://designateddriver.ai/. Accessed on 03.04.2021.

[47] Marius Dupuis et al. "Opendrive format specification". In: *VIRES Simulationstechnologie GmbH* (2010).

[48] Henrikke Dybvik, Martin Løland, Achim Gerstenberg, Kristoffer Bjørnerud Slattsveen, and Martin Steinert. "A low-cost predictive display for teleoperation: Investigating effects on human performance and workload". In: *International Journal of Human-Computer Studies* 145 (2021), p. 102536. ISSN: 1071-5819. DOI: https://doi.org/10.1016/j.ijhcs.2020.102536. URL: https://www.sciencedirect.com/science/article/pii/S1071581920301385.

[49] Einride. *Einride, intelligent technologies for movement - Einride*. https://www.einride.tech/. Accessed on 25.05.2021.

[50] Elektrobit. *EB Assist ADTF - Elektrobit*. https://www.elektrobit.com/products/automated-driving/eb-assist/adtf/. Accessed on 10.04.2019.

[51] Ericsson. *Remote operation of vehicles with 5G*. https://www.ericsson.com/4add9b/assets/local/mobility-report/documents/2017/emr-november-2017-remote-operation-of-vehicles-with-5g.pdf. Accessed on 10.03.2020. June 2017.

[52] Euro NCAP. *Offizielle Sicherheitsbewertung BMW 3 Series 2019*. https://www.euroncap.com/de/results/bmw/3-series/38531. Accessed on 27.12.2020.

[53] Eurolink Systems. *Leopardo B NATO project*. https://www.youtube.com/watch?v=g3zuAb127Rg. Accessed on 14.09.2022. Apr. 2021.

[54] Francesca M. Favarò, Nazanin Nader, Sky O. Eurich, Michelle Tripp, and Naresh Varadaraju. "Examining accident reports involving autonomous vehicles in California". In: *PLOS ONE* 12.9 (Sept. 2017), pp. 1–20. DOI: 10.1371/journal.pone.0184952. URL: https://doi.org/10.1371/journal.pone.0184952.

[55] Federal Ministry for Digital and Transport. *Germany will be the world leader in autonomous driving*. https://bmdv.bund.de/SharedDocs/EN/Articles/DG/act-on-autonomous-driving.html. Accessed: 22.12.2022. July 2021.

[56] Federal Ministry of Transport and Digital Infrastructure. *A scheme to promote the development of Germany to become a lead market for 5G networks and applications*. Tech. rep. Berlin, Germany: Federal Ministry of Transport and Digital Infrastructure, July 2017.

[57] Fernride GmbH. *Fernride - Enabling driverless logistics*. https://www.fernride.com/. Accessed on 03.04.2021.

[58] FFmpeg Developers. *FFmpeg*. https://www.ffmpeg.org/. Accessed on 20.12.2019. 2019.

[59] A. Gaber, W. Nassar, A. M. Mohamed, and M. K. Mansour. “Feasibility Study of Teleoperated Vehicles Using Multi-Operator LTE Connection”. In: *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*. 2020, pp. 191–195. DOI: 10.1109/ITCE48509.2020.9047764.

[60] Douglas W Gage. *UGV history 101: A brief history of Unmanned Ground Vehicle (UGV) development efforts*. Tech. rep. NAVAL COMMAND CONTROL, OCEAN SURVEILLANCE CENTER RDT, and E DIV SAN DIEGO CA, 1995.

[61] Luis Govinda García-Valdovinos, Tomás Salgado-Jiménez, Manuel Bandala-Sánchez, Luciano Nava-Balanzar, Rodrigo Hernández-Alvarado, and José Antonio Cruz-Ledesma. “Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle”. In: *International Journal of Advanced Robotic Systems* 11.1 (2014), p. 1. DOI: 10.5772/56810. eprint: https://doi.org/10.5772/56810. URL: https://doi.org/10.5772/56810.

[62] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* (2013).

[63] J. Georg, J. Feiler, F. Diermeyer, and M. Lienkamp. “Teleoperated Driving, a Key Technology for Automated Driving? Comparison of Actual Test Drives with a Head Mounted Display and Conventional Monitors*”. In: *21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI, USA: IEEE, Nov. 2018, pp. 3403–3408. DOI: 10.1109/ITSC.2018.8569408.

[64] J. -M. Georg, J. Feiler, S. Hoffmann, and F. Diermeyer. “Sensor and Actuator Latency during Teleoperation of Automated Vehicles”. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA: IEEE, Nov. 2020, pp. 760–766. DOI: 10.1109/IV47402.2020.9304802.

[65] Jean-Michael Georg, Elena Putz, and Frank Diermeyer. “Longtime Effects of Videoquality, Videocanvases and Displays on Situation Awareness during Teleoperation of Automated Vehicles*”. In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Toronto, ON, Canada: IEEE, Oct. 2020, pp. 248–255. DOI: 10.1109/SMC42975.2020.9283364.

[66] German Aerospace Center (DLR). *TraCI*. https://sumo.dlr.de/docs/TraCI.html. Accessed on 03.04.2020. Mar. 2020. URL: https://sumo.dlr.de/docs/TraCI.html.

[67] Geyer, Jakob and others. *A2D2: AEV Autonomous Driving Dataset*. http://www.a2d2.audi. 2019.

[68] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. "A System Design for Teleoperated Road Vehicles." In: *ICINCO 2013 - Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*. Reykjavik; Iceland, July 2013, pp. 231–238.

[69] J. Gozalvez. "Samsung Electronics Sets 5G Speed Record at 7.5 Gbs [Mobile Radio]". In: *IEEE Vehicular Technology Magazine* 10.1 (2015), pp. 12–16.

[70] G. Graf, H. Xu, D. Schitz, and X. Xu. "Improving the Prediction Accuracy of Predictive Displays for Teleoperated Autonomous Vehicles". In: *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. 2020, pp. 440–445. DOI: 10.1109/ICCAR49639.2020.9108011.

[71] Friedhelm Greis. *Per Fernsteuerung durch die Baustelle*. https://www.golem.de/news/autonomes-fahren-per-fernsteuerung-durch-die-baustelle-1906-141791.html. Accessed on 22.11.2020. June 2019.

[72] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar. "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders". In: *2013 Picture Coding Symposium (PCS)*. Dec. 2013, pp. 394–397. DOI: 10.1109/PCS.2013.6737766.

[73] M. Guimaraes and J. Lindberg. "Remote controlled vehicle for inspection of vertical concrete structures". In: *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*. 2014, pp. 1–6. DOI: 10.1109/CARPI.2014.7030048.

[74] Hang Guo and John Heidemann. "Detecting ICMP Rate Limiting in the Internet". In: *Passive and Active Measurement*. Ed. by Robert Beverly, Georgios Smaragdakis,

and Anja Feldmann. Cham: Springer International Publishing, 2018, pp. 3–17. ISBN: 978-3-319-76481-8.

[75] Mark Harris. *CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads*. https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads. Accessed on 28.11.2018. Jan. 2018.

[76] Sandra G. Hart and Lowell E. Staveland. “Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research”. In: *Human Mental Workload*. Ed. by Peter A. Hancock and Najmedin Meshkati. Vol. 52. Advances in Psychology. North-Holland, 1988, pp. 139–183. DOI: https://doi.org/10.1016/S0166-4115(08)62386-9. URL: http://www.sciencedirect.com/science/article/pii/S0166411508623869.

[77] Andrew J. Hawkins. *Drone truck startup Einride unveils new driverless vehicles for autonomous freight hauling*. https://www.theverge.com/2020/10/8/21506125/einride-self-driving-truck-pod-aet-reveal. Accessed on 25.05.2021. Oct. 2020.

[78] Andrew J. Hawkins. *Waymo pulls back the curtain on 6.1 million miles of self-driving car data in Phoenix*. https://www.theverge.com/2020/10/30/21538999/waymo-self-driving-car-data-miles-crashes-phoenix-google. Accessed on 27.12.2020. Oct. 2020.

[79] Andrew J. Hawkins. *Waymo will allow more people to ride in its fully driverless vehicles in Phoenix*. https://www.theverge.com/2020/10/8/21507814/waymo-driverless-cars-allow-more-customers-phoenix. Accessed on 27.12.2020. Oct. 2020.

[80] Gregor Hebermehl and Uli Baumann. *Mercedes Drive Pilot: Autonom nach Level 3 — AUTO MOTOR UND SPORT*. https://www.auto-motor-und-sport.de/tech-zukunft/mercedes-autonom-level-3-drive-pilot-haftung-unfall/. Accessed on 07.09.2022. May 2022.

[81] Rebecca Heilweil. *Tesla needs to fix its deadly Autopilot problem*. https://www.vox.com/recode/2020/2/26/21154502/tesla-autopilot-fatal-crashes. Accessed on 27.12.2020. Feb. 2020.

[82] Stephen B. Heppe. "Problem of UAV Communications". In: *Handbook of Unmanned Aerial Vehicles*. Ed. by Kimon P. Valavanis and George J. Vachtsevanos. Dordrecht: Springer Netherlands, 2015, pp. 715–748. ISBN: 978-90-481-9707-1. DOI: 10.1007/978-90-481-9707-1_30. URL: https://doi.org/10.1007/978-90-481-9707-1_30.

[83] M. Hofbauer, C. B. Kuhn, G. Petrovic, and E. Steinbach. "Adaptive Multi-View Live Video Streaming for Teledriving Using a Single Hardware Encoder". In: *2020 IEEE International Symposium on Multimedia (ISM)*. 2020, pp. 9–16. DOI: 10.1109/ISM.2020.00008.

[84] Markus Hofbauer, Christopher B. Kuhn, Mariem Khlifi, Goran Petrovic, and Eckehard Steinbach. "Traffic-Aware Multi-View Video Stream Adaptation for Teleoperated Driving". In: *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*. Helsinki, Finland: IEEE, Aug. 2022, pp. 1–7. DOI: 10.1109/VTC2022-Spring54318.2022.9860513.

[85] A. Hosseini and M. Lienkamp. "Enhancing telepresence during the teleoperation of road vehicles using HMD-based mixed reality". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. Gothenburg, Sweden: IEEE, June 2016, pp. 1366–1373. DOI: 10.1109/IVS.2016.7535568.

[86] A. Hosseini and M. Lienkamp. "Predictive safety based on track-before-detect for teleoperated driving through communication time delay". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. June 2016, pp. 165–172. DOI: 10.1109/IVS.2016.7535381.

[87] A. Hosseini, F. Richthammer, and M. Lienkamp. "Predictive Haptic Feedback for Safe Lateral Control of Teleoperated Road Vehicles in Urban Areas". In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. Nanjing, China: IEEE, May 2016, pp. 1–7. DOI: 10.1109/VTCSpring.2016.7504430.

[88] Hui-Min Huang, Kerry Pavek, James Albus, and Elena Messina. "Autonomy levels for unmanned systems (ALFUS) framework: An update". In: *Unmanned Ground Vehicle Technology VII*. Vol. 5804. International Society for Optics and Photonics. 2005, pp. 439–448.

[89] R. Inam, N. Schrammar, K. Wang, A. Karapantelakis, L. Mokrushin, A. V. Feljan, and E. Fersman. "Feasibility assessment to realise vehicle teleoperation using cellular networks". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016, pp. 2254–2260. DOI: 10.1109/ITSC.2016.7795920.

[90] Intuitive Surgical. *About da Vinci Systems*. https://www.davincisurgery.com/da-vinci-systems/about-da-vinci-systems. Accessed on 29.12.2020. Mar. 2019.

[91] H. Z. Jahromi, I. Bartolec, E. Gamboa, A. Hines, and R. Schatz. "You Drive Me Crazy! Interactive QoE Assessment for Telepresence Robot Control". In: *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. 2020, pp. 1–6. DOI: 10.1109/QoMEX48832.2020.9123117.

[92] Keon Jang, Mongnam Han, Soohyun Cho, Hyung-Keun Ryu, Jaehwa Lee, Yeongseok Lee, and Sue B. Moon. "3G and 3.5G Wireless Network Performance Measured from Moving Cars and High-speed Trains". In: *Proceedings of the 1st ACM Workshop on Mobile Internet Through Cellular Networks*. MICNET '09. Beijing, China: ACM, 2009, pp. 19–24. ISBN: 978-1-60558-753-0. DOI: 10.1145/1614255.1614261. URL: http://doi.acm.org/10.1145/1614255.1614261.

[93] Lei Kang, Wei Zhao, Bozhao Qi, and Suman Banerjee. "Augmenting Self-Driving with Remote Control: Challenges and Directions". In: *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*. HotMobile '18. Tempe, Arizona, USA: ACM, 2018, pp. 19–24. ISBN: 978-1-4503-5630-5. DOI: 10.1145/3177102.3177104. URL: http://doi.acm.org/10.1145/3177102.3177104.

[94] Charles F. F. Karney. "Algorithms for geodesics". In: *Journal of Geodesy* 87.1 (Jan. 2013), pp. 43–55. ISSN: 1432-1394. DOI: 10.1007/s00190-012-0578-z. URL: https://doi.org/10.1007/s00190-012-0578-z.

[95] R. Kesten et al. *Lyft Level 5 AV Dataset 2019*. https://level5.lyft.com/dataset/. 2019.

[96] khanhlvg, tombstone, a-googler, srjoglekar246, and pkulzc. *TensorFlow 2 Detection Model Zoo*. https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md. Accessed: 10.11.2020. 2020.

[97] Seiga Kiribayashi, Kaede Yakushigawa, and Keiji Nagatani. "Design and Development of Tether-Powered Multirotor Micro Unmanned Aerial Vehicle System for Remote-Controlled Construction Machine". In: *Field and Service Robotics*. Ed. by Marco Hutter and Roland Siegwart. Cham: Springer International Publishing, 2018, pp. 637–648. ISBN: 978-3-319-67361-5.

[98] G. Knappe, A. Keinath, K. Bengler, and C. Meinecke. "Driving Simulators as an Evaluation Tool - Assessment of the Influence of the Field of View and Secondary Tasks on Lane Keeping and Steering Performance". In: *20th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*. Vol. 810-736. DOT-HS. Lyon , France, 2007.

[99] Gwendolin Knappe. "Empirische Untersuchungen zur Querregelung in Fahrsimulatoren - Vergleichbarkeit von Untersuchungsergebnissen und Sensitivität von Messgrößen". PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, Nov. 2009.

[100] Panos Konstantopoulos, Peter Chapman, and David Crundall. "Driver's visual attention as a function of driving experience and visibility. Using a driving simulator to explore drivers' eye movements in day, night and rain driving". In: *Accident Analysis & Prevention* 42.3 (2010), pp. 827–834.

[101] Kirsten Korosec. *Starship Technologies is sending its autonomous robots to more cities as demand for contactless delivery rises*. https://techcrunch.com/2020/04/09/starship-technologies-is-sending-its-autonomous-robots-to-more-cities-as-demand-for-contactless-delivery-rises. Accessed: 21.01.2021. Apr. 2020.

[102] Andrew Krok. *GM's Super Cruise aims to be in 22 vehicles by 2023*. https://www.cnet.com/roadshow/news/gm-super-cruise-expansion-2023/. Accessed on 27.12.2020. Feb. 2020.

[103] Jonathan Kua, Grenville Armitage, and Philip Branch. "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP". In: *IEEE Communications Surveys & Tutorials* 19.3 (Mar. 2017), pp. 1842–1866. DOI: 10.1109/COMST.2017.2685630.

[104] Kristofer D. Kusano and Hampton C. Gabler. "Safety Benefits of Forward Collision Warning, Brake Assist, and Autonomous Braking Systems in Rear-End Collisions". In: *IEEE Transactions on Intelligent Transportation Systems* 13.4 (Apr. 2012), pp. 1546–1555. DOI: 10.1109/TITS.2012.2191542.

[105] Marten Lager and Elin A. Topp. "Remote Supervision of an Autonomous Surface Vehicle using Virtual Reality". In: *IFAC-PapersOnLine* 52.8 (2019). 10th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2019, pp. 387–392. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2019.08.104. URL: https://www.sciencedirect.com/science/article/pii/S2405896319304380.

[106] Emily Lakdawalla. *The design and engineering of Curiosity: how the Mars Rover performs its job*. Springer, 2018.

[107] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen. "From LTE to 5G for Connected Mobility". In: *IEEE Communications Magazine* 55.3 (Mar. 2017), pp. 156–162. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600778CM.

[108] Dave Lee. *Uber self-driving crash 'mostly caused by human error*. https://www.bbc.com/news/technology-50484172. Accessed on 27.12.2020. Nov. 2019.

[109] Dominik J Leiner. "Our research's breadth lives on convenience samples A case study of the online respondent pool "SoSci Panel"". In: *SCM Studies in Communication and Media* 5.4 (2017), pp. 367–396.

[110] Lenovo. *Lenovo B smartphone*. https://www.lenovo.com/gb/en/smart-devices/smartphones-and-watches/lenovo/lenovo-series/Lenovo-Vibe-B/p/PPIPPIA2016. Accessed on 27.09.2018.

[111] Feng Li, Jae Won Chung, Xiaoxiao Jiang, and Mark Claypool. "TCP CUBIC versus BBR on the Highway". In: *Passive and Active Measurement*. Ed. by Robert Beverly, Georgios Smaragdakis, and Anja Feldmann. Cham: Springer International Publishing, 2018, pp. 269–280. ISBN: 978-3-319-76481-8.

[112] Zhi Li et al. *Toward A Practical Perceptual Video Quality Metric*. https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652. Accessed on 18.12.2019. June 2016.

[113] S Lichiardopol. "A survey on teleoperation". In: *Technische Universitat Eindhoven, DCT report* 20 (2007), pp. 40–60.

[114] Todd Litman. *Autonomous Vehicle Implementation Predictions - Implications for Transport Planning*. Tech. rep. Victoria Transport Policy Institute Victoria, Canada, Mar. 2019.

[115] Ruilin Liu, Daehan Kwak, Srinivas Devarakonda, Kostas Bekris, and Liviu Iftode. "Investigating Remote Driving over the LTE Network". In: *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '17. Oldenburg, Germany: ACM, 2017, pp. 264–269. ISBN: 978-1-4503-5150-8. DOI: 10.1145/3122986.3123008. URL: http://doi.acm.org/10.1145/3122986.3123008.

[116] Waymo LLC. *About – Waymo*. https://waymo.com/open/about/. Accessed on 08.01.2020.

[117] Logitech. *Logitech G920 & G29 Driving Force Steering Wheels & Pedals*. https://www.logitechg.com/en-us/products/driving/driving-force-racing-wheel.html. Accessed: 08.07.2019. 2019.

[118] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. Maui, HI, USA: IEEE, 2018. URL: https://elib.dlr.de/124092/.

[119] Jason P. Luck, Patricia L. McDermott, Laurel Allender, and Deborah C. Russell. "An Investigation of Real World Control of Robotic Assets Under Communication Latency". In: *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*. HRI '06. Salt Lake City, Utah, USA: ACM, 2006, pp. 202–209. ISBN: 1-59593-294-1. DOI: 10.1145/1121241.1121277. URL: http://doi.acm.org/10.1145/1121241.1121277.

[120] Aarian Marshall. *Self-Driving Shuttle Buses Might Be the Future of Transportation*. https://www.wired.com/story/las-vegas-shuttle-crash-self-driving-autonomous/. Accessed on 27.12.2020. Oct. 2017.

[121] Rafael Math, Angela Mahr, Mohammad M Moniri, and Christian Müller. "OpenDS: A new open-source driving simulator for research". In: *AmE 2013 - Automotive meets Electronics*. 2013.

[122] Mercedes Benz. *Mercedes-Benz S-Class: Intelligent Drive Next Level*. https://wwwmercedes-benz.com/en/innovation/autonomous/the-new-s-class-intelligent-drive-next-level/. Accessed on 27.12.2020. Apr. 2017.

[123] Mercedes-Benz Group AG. *The front runner in automated driving and safety technologies*. https://group.mercedes-benz.com/innovation/case/autonomous/drive-pilot-2.html. Accessed on 03.09.2022. Apr. 2022.

[124] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. "Performance of LTE in a High-velocity Environment: A Measurement Study". In: *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges*. AllThingsCellular '14. Chicago, Illinois, USA: ACM, 2014, pp. 47–52. ISBN: 978-1-4503-2990-3. DOI: 10.1145/2627585.2627589. URL: http://doi.acm.org/10.1145/2627585.2627589.

[125] Manfred Mitschke and Henning Wallentowitz. "Kreisfahrt bei konstanter Fahrgeschwindigkeit". In: *Dynamik der Kraftfahrzeuge*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014, pp. 625–655. ISBN: 978-3-658-05068-9. DOI: 10.1007/978-3-658-05068-9_21. URL: https://doi.org/10.1007/978-3-658-05068-9_21.

[126] N. Mouawad, R. Naja, and S. Tohme. "SDN Based Handover Management For a Tele-Operated Driving Use Case". In: *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*. 2019, pp. 47–54. DOI: 10.23919/WMNC.2019.8881405.

[127] MulticoreWare Inc. *Command Line Options - x265 documentation*. https://x265.readthedocs.io/en/stable/cli.html. Accessed on 11.10.2020.

[128] NASA. *Autonomous Planetary Mobility*. https://mars.nasa.gov/mer/mission/technology/autonomous-planetary-mobility/. Accessed on 27.12.2020.

[129] NASA. *Communications - NASA Mars*. https://mars.nasa.gov/mars2020/spacecraft/rover/communications/. Accessed on 25.05.2021. Mar. 2021.

[130] NASA. *Communications with Earth — Mission – NASA's Mars Exploration Program*. https://mars.nasa.gov/msl/mission/communications/. Accessed on 27.12.2020. Aug. 2019.

[131] NASA. *Summary — Rover – NASA's Mars Exploration Program*. https://mars.nasa.gov/msl/spacecraft/rover/summary/. Accessed on 27.12.2020. Nov. 2020.

[132] NASA/JPL-Caltech/MSSS. *PIA19920: Curiosity Self-Portrait at 'Big Sky' Drilling Site*. https://photojournal.jpl.nasa.gov/catalog/PIA19920. Accessed on 28.12.2020. Oct. 2015.

[133] Rajkishore Nayak, Rajiv Padhye, Sinnappoo Kanesalingam, Lyndon Arnold, and Bijoy Kumar Behera. "Airbags". In: *Textile Progress* 45 (Dec. 2013), pp. 209–301. DOI: 10.1080/00405167.2013.859435.

[134] Next Level Racing. *Next Level Racing Motion Platform V3 - Next Level Racing*. https://nextlevelracing.com/products/next-level-racing-motion-platform-v3/. Accessed on 07.01.2022. 2023.

[135] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks. "Ecological Interfaces for Improving Mobile Robot Teleoperation". In: *IEEE Transactions on Robotics* 23.5 (Oct. 2007), pp. 927–941. ISSN: 1552-3098. DOI: 10.1109/TRO.2007.907479.

[136] M. Z. H. Noor, S. A. S. M. Zain, and L. Mazalan. “Design and development of remote-operated multi-direction Unmanned Ground Vehicle (UGV)”. In: *2013 IEEE 3rd International Conference on System Engineering and Technology*. 2013, pp. 188–192. DOI: 10.1109/ICSEngT.2013.6650168.

[137] OpenCelliD. *OpenCelliD - Largest Open Database of Cell Towers & Geolocation - by Unwired Labs*. https://opencellid.org/. Accessed on 10.03.2019.

[138] OpenCV. *Image Filtering - Image Processing*. https://docs.opencv.org/4.2.0/d4/d86/group__imgproc__filter.html. Accessed: 10.11.2020. 2020.

[139] OpenStreetMap contributors. *JOSM*. https://josm.openstreetmap.de/. Accessed on 04.04.2019.

[140] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2017.

[141] Ordercrazy. *Embedded World 2014: Unmanned Ground Vehicle (UGV) Robot Leopardo B, built by EuroLink Systems shown at Embedded World Fair 2014 in Nuremberg*. https://commons.wikimedia.org/wiki/File:Embedded_World_Leopardo_2_UGV_01_(02).jpg. Accessed on 29.12.2020. Feb. 2014.

[142] A. Parichehreh, S. Alfredsson, and A. Brunstrom. “Measurement Analysis of TCP Congestion Control Algorithms in LTE Uplink”. In: *Traffic Measurement and Analysis Conference*. June 2018, pp. 1–8. DOI: 10.23919/TMA.2018.8506522.

[143] Sylvain Paris. “A Gentle Introduction to Bilateral Filtering and Its Applications”. In: *ACM SIGGRAPH 2007 Courses*. SIGGRAPH ’07. San Diego, California: Association for Computing Machinery, 2007, 3–es. ISBN: 9781450318235. DOI: 10.1145/1281500.1281604. URL: https://doi.org/10.1145/1281500.1281604.

[144] Phantom Auto. *Phantom Auto - Enabling Autonomy Through Teleoperation*. https://phantom.auto/. Accessed on 03.04.2021.

[145] *ping utility for Android*. https://bit.ly/2Hr4Tzi. 2019.

[146] Craig Quiter and Maik Ernst. *deepdrive/deepdrive: 2.0*. Mar. 2018. DOI: 10.5281/zenodo.1248998. URL: https://doi.org/10.5281/zenodo.1248998.

[147] Justus Rischke, Peter Sossalla, Sebastian Itting, Frank H. P. Fitzek, and Martin Reisslein. "5G Campus Networks: A First Measurement Study". In: *IEEE Access* 9 (Aug. 2021), pp. 121786–121803. DOI: 10.1109/ACCESS.2021.3108423.

[148] On-Road Automated Driving (ORAD) committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International, June 2018.

[149] Anirban Santara. *MADRaS: A Multi-Agent DRiving Simulator*. https://software.intel.com/en-us/articles/madras-a-multi-agent-driving-simulator. Accessed on 10.04.2019. Apr. 2018.

[150] Martin Sauter. *From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband*. Third. WILEY, 2017. ISBN: 9781119346869.

[151] A. Schimpe and F. Diermeyer. "Steer with Me: A Predictive, Potential Field-Based Control Approach for Semi-Autonomous, Teleoperated Road Vehicles". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294702.

[152] Andreas Schimpe, Simon Hoffmann, and Frank Diermeyer. *Adaptive Video Configuration and Bitrate Allocation for Teleoperated Driving*. 2021. arXiv: 2102.10898 [eess.IV].

[153] D. Schitz, G. Graf, D. Rieth, and H. Aschemann. "Interactive Corridor-Based Path Planning for Teleoperated Driving". In: *2021 7th International Conference on Mechatronics and Robotics Engineering (ICMRE)*. 2021, pp. 174–179. DOI: 10.1109/ICMRE51691.2021.9384848.

[154] Dieter Schramm, Manfred Hiller, and Roberto Bardini. "Single Track Models". In: *Vehicle Dynamics: Modeling and Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 223–253. ISBN: 978-3-540-36045-2. DOI: 10.1007/978-3-540-36045-2_10. URL: https://doi.org/10.1007/978-3-540-36045-2_10.

[155] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles". In: *Field and Service Robotics*. 2017. eprint: `arXiv:1705.05065`. URL: `https://arxiv.org/abs/1705.05065`.

[156] Xiaotong Shen, Zhuang Jie Chong, Scott Pendleton, Guo Ming James Fu, Baoxing Qin, Emilio Frazzoli, and Marcelo H. Ang. "Teleoperation of On-Road Vehicles via Immersive Telepresence Using Off-the-shelf Components". In: *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*. Ed. by Emanuele Menegatti, Nathan Michael, Karsten Berns, and Hiroaki Yamaguchi. Cham: Springer International Publishing, 2016, pp. 1419–1433. ISBN: 978-3-319-08338-4. DOI: `10.1007/978-3-319-08338-4\_102`. URL: `http://dx.doi.org/10.1007/978-3-319-08338-4%5C_102`.

[157] T. B. Sheridan. "Space teleoperation through time delay: review and prognosis". In: *IEEE Transactions on Robotics and Automation* 9.5 (Oct. 1993), pp. 592–606. ISSN: 1042-296X. DOI: `10.1109/70.258052`.

[158] K. Shuaib, T. Saadawi, M. Lee, and B. Basch. "De-jittering in the transport of MPEG-4 and MPEG-2 video over ATM". In: *1999 2nd International Conference on ATM. ICATM'99 (Cat. No.99EX284)*. Colmar, France: IEEE, Aug. 1999, pp. 502–506. DOI: `10.1109/ICATM.1999.786845`.

[159] SierraWireless. *AirLink RV50 Datasheet*. `https://www.sierrawireless.com/-/media/iot/pdf/datasheets/sierrawireless_airlink_rv50_datasheet.pdf`. Accessed on 27.09.2018.

[160] Matt Simon. *Curiosity Doesn't Need Your Help Blasting Rocks With a Laser*. `https://www.wired.com/story/curiosity-rover-autonomy/`. Accessed on 27.12.2020. June 2017.

[161] Werner Spielmann and Manfred Reuter. "Bremswege als Vergleichsgröße zwischen Bremsanlagen". In: *ATZ - Automobiltechnische Zeitschrift* 104.5 (May 2002), pp. 464–472. ISSN: 2192-8800. DOI: `10.1007/BF03224409`. URL: `https://doi.org/10.1007/BF03224409`.

[162] NATO STANAG. “4671 Unmanned Aerial Vehicles Systems Airworthiness Requirements (USAR)”. In: *NSA/0976* (2009).

[163] STARSKY ROBOTICS. *Starsky Robotics - Bringing Driverless Trucks to Market*. https://www.starsky.io/. Accessed on 03.04.2021.

[164] Yu-Chuan Su and Kristen Grauman. “Learning Compressible 360° Video Isomers”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[165] Tito Tang, Frederic Chucholowski, and Markus Lienkamp. “Teleoperated driving basics and system design”. In: *ATZ worldwide* 116.2 (Feb. 2014), pp. 16–19. ISSN: 2192-9076. DOI: 10.1007/s38311-014-0018-1. URL: https://doi.org/10.1007/s38311-014-0018-1.

[166] Tito Tang, Pascal Vetter, Markus Lienkamp, Simon Finkl, and Korbinian Figel. “Teleoperated Road Vehicles – The ”Free Corridor” as a Safety Strategy Approach”. In: *Mechanical Design and Power Engineering*. Vol. 490. Applied Mechanics and Materials. Trans Tech Publications, Apr. 2014, pp. 1399–1409. DOI: 10.4028/www.scientific.net/AMM.490-491.1399.

[167] Unity Technologies. *Windridge City*. https://assetstore.unity.com/packages/3d/environments/roadways/windridge-city-132222. Accessed on 05.03.2019. Jan. 2019.

[168] M. Toril, V. Wille, S. Luna-Ramírez, M. Fernández-Navarro, and F. Ruiz-Vega. “Characterization of Radio Signal Strength Fluctuations in Road Scenarios for Cellular Vehicular Network Planning in LTE”. In: *IEEE Access* 9 (2021), pp. 33120–33131. DOI: 10.1109/ACCESS.2021.3060995.

[169] Udacity. *GitHub - udacity/self-driving-car: The Udacity open source self-driving car project*. https://github.com/udacity/self-driving-car. Accessed on 08.01.2020.

[170] Udacity. *Udacity’s Self-Driving Car Simulator*. https://github.com/udacity/self-driving-car-sim. Accessed on 10.04.2019.

[171] Unity Technologies. *Unity*. https://unity.com/. Accessed on 11.04.2019.

[172] Vector Informatik GmbH. *DYNA4 - Virtual Test Driving*. https://www.vector.com/int/en/products/products-a-z/software/dyna4. Accessed on 30.09.2022.

[173] Steve Vozar and Dawn M. Tilbury. "Driver Modeling for Teleoperation with Time Delay". In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 3551–3556. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20140824-6-ZA-1003.02275. URL: http://www.sciencedirect.com/science/article/pii/S1474667016421555.

[174] Jeanette Wannstrom. "LTE-advanced". In: *Third Generation Partnership Project (3GPP)* (2012).

[175] Waymo. *Waymo Open Dataset: An autonomous driving dataset*. https://www.waymo.com/open. 2019.

[176] Alan Ft Winfield. "Future directions in tele-operated robotics". In: *Telerobotic applications* (2000), pp. 147–163.

[177] Working Group 1. *Tele-Operated Driving (ToD): Use Cases and Technical Requirements*. Tech. rep. Munich, Germany: 5GAA Automotive Association, July 2020.

[178] Wuerzburger Institut fuer Verkehrswissenschaften GmbH. *Driving Simulation and SILAB*. https://wivw.de/en/silab. Accessed on 30.09.2022.

[179] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. *Torcs, the open racing car simulator*. Tech. rep. Mar. 2015.

[180] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong. "TCP Performance over Mobile Networks in High-Speed Mobility Scenarios". In: *2014 IEEE 22nd International Conference on Network Protocols*. Oct. 2014, pp. 281–286. DOI: 10.1109/ICNP.2014.49.

[181] Y. Yu and S. Lee. "A Measurement Based Feasibility Study on Remote Driving System". In: *2021 International Conference on Information Networking (ICOIN)*. 2021, pp. 17–19. DOI: 10.1109/ICOIN50884.2021.9333987.

Appendix - Publications

Publication [VII]: Measuring the Feasibility of Teleoperated Driving in Mobile Networks

[**VII**] Stefan Neumeier, Ermias Walelgne, Vaibhav Bajpai, Joerg Ott, and Christian Facchi. "Measuring the Feasibility of Teleoperated Driving in Mobile Networks". In: *2019 Network Traffic Measurement and Analysis Conference (TMA)*. Paris, France: IEEE, June 2019, pp. 113–120

This paper puts the focus on real-world measurements in order to check whether contemporary cellular networks are suitable for the needs of teleoperated driving. Therefore, we condcuted cellular network measurements while driving a car in Germany. Measurements addressed uplink and downlink bandwidth, latency and network coverage, which are all important factors for teleoperated driving. The covered distance for the measurements was about 5200 km, which is related to approximately 80 h of driving. In order to produce comparable results, we utilized different hardware and software setups. The hardware was either a LTE-Modem fixed to the trunk of a test vehicle or a typical smartphone. For measurements with the dedicated LTE-Modem a setup with iperf3 was utilized. In contrast, the smartphone was running a ping application and netradar. We analyzed the collected data with focus on values that are required for teleoperated driving, i. e., it was calculated how often the uplink speed and downlink speed was below/above the defined bandwidth requirements. We did the same analysis for latency and the corresponding jitter. Results indicated that teleoperated driving could be used in the majority of the measured cases, e. g., depending on the measurement setup, at least 87% of the measurements were above the minimal value required for uplink, for latency about 96% of the measurements were below the critical value. It was further analyzed whether factors such as handover, speed, signal strength, distance to a base station and distance to the server have an impact on the measured network performance. Luckily it turned out that the speed of the vehicle and the distance to the base station have no negative influence on the network performance. However, handover, distance to the server and signal strength can have an impact on the network performance. One method to overcome the issues induced by cellular networks is the use of a whitelist approach, i. e., remote vehicles are only allowed for remote operation in areas that are known to provide sufficient cellular network performance. We conducted measurements in an area, which was selected to check whether whitelisting could work in practice. Results indicated that a whitelist approach could help to keep vehicles in areas with sufficient cellular network performance.

The initial idea of this paper was developed by me together with the co-authors. Most parts of the self-developed used software, excluding netradar, was developed by working students based on the requirements I specified. The servers were also set up by me where required. The driving to conduct the measurements was done by my, working students and supporting colleagues. The setup for the whitelisting approach was done by a student, but supported by me. The analysis and interpretation of the results was mainly done by me, but with a close exchange with the co-authors, who also provided directions to further investigate. The writing itself was mainly done by me with support of the co-authors. I also mainly took care of addressing reviewers' comments.

Copyright Clearance Center RightsLink® Home Help Email Support Sign in Create Account

IEEE Requesting permission to reuse content from an IEEE publication

Measuring the Feasibility of Teleoperated Driving in Mobile Networks

Conference Proceedings: 2019 Network Traffic Measurement and Analysis Conference (TMA)

Author:
[::Stefan::] [::Neumeier::]; Ermias Andargie Walelgne; Vaibhav Bajpai; Jörg Ott; Christian Facchi

Publisher: IEEE

Date: 19-21 June 2019

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Measuring the Feasibility of Teleoperated Driving in Mobile Networks

Stefan Neumeier*, Ermias Andargie Walelgne‡, Vaibhav Bajpai†, Jörg Ott† and Christian Facchi*
*Technische Hochschule Ingolstadt, Research Centre
(stefan.neumeier | christian.facchi)@thi.de
†Technische Universität München
(bajpaiv | ott)@in.tum.de
‡Aalto University
ermias.walelgne@aalto.fi

***Abstract*—Teleoperated Driving is the remote control driving of a vehicle by a human driver. The concept of Teleoperated Driving requires the use of mobile networks, which typically experience variable throughput, variable latency and uneven network coverage. To investigate whether Teleoperated Driving can be possible with contemporary mobile networks, we have conducted measurements while driving with vehicles in the real world. We used complementary measurement setups to obtain results that can be compared. The dataset consists of about 5200 km (4660 minutes) driving measurements. Results show that Teleoperated Driving could be possible, but the high variance of network parameters makes it difficult to use the system at all times. It appears that the speed of the vehicle and the distance to the base station may not influence Teleoperated Driving, while handover with changed radio technology, signal strength and distance to the teleoperation station may have an impact. Possible mitigations to overcome these problems along with a basic whitelisting approach is discussed.**

I. Introduction

The Working Group "Connectivity and Automated Driving" of the European Road Transport Research Advisory Council (ERTRAC) [11], a technology platform developing a common vision for road transport in Europe, show that vehicles will advance in autonomous driving features until they are fully autonomous. However, until fully autonomous vehicles are available, there can be situations where autonomous features would be useful but not existent. A solution to provide autonomous-like behavior of vehicles is Teleoperated Driving, where a vehicle is controlled remotely by a human driver when required. Even with fully autonomous vehicles, there will be situations where a system can not handle a situation and human intervention is necessary, e.g. complex road side work [17] or other obstacles [20]. In such a scenario, the remote operator takes over control and operates the vehicle as long as required, typically covering short distances. To safely control a remote vehicle in traffic, it is important that the teleoperator monitors the environment of the remote vehicle constantly and is able to deal with different situations. This can be difficult due to the non-deterministic behavior of the utilized mobile network. Sufficient bidirectional throughput is required to exchange data between the remote vehicle and the teleoperation station. One direction is required for the remote vehicle to provide the driver with environmental information, whereas the other direction is used for transmitting steering commands. Further challenges are latency [31] and jitter. Teleoperated vehicles use regular streets and thus have to deal with suddenly appearing obstacles. Latency is one of the key indicators to determine how safely one can control the vehicle, as transmission of steering commands and streams can get delayed. Due to the mobility of vehicles, a high frequency in the changes of network conditions can be expected [34]. This makes Teleoperated Driving even more challenging. Considering these obstacles, we want to know: Is Teleoperated Driving feasible with contemporary mobile networks? To answer this question the paper provides a first assessment on this topic. We investigate latency and throughput values while driving in the real world using three complementary measurement setups. We assess, whether factors such as handover, distance of a remote car to the teleoperation station or signal strength have an influence on the usability of the system. We show that Teleoperated Driving can be feasible and observe that signal strength, the distance to the teleoperation station and handover with changed radio technology can have an influence on Teleoperated Driving, while the speed of the vehicle and distance to base station do not have an influence. We further investigate whether an approach of white listing areas with good network conditions can help Teleoperated Driving.

The paper is structured as follows. Section II provides an overview of related work. Section III defines the network requirements for Teleoperated Driving. Section IV introduces the measurement setup and the collected dataset. Section V presents the results, discusses the influence of different parameters and proposes mitigations. Section VI discusses limitations and future work. Finally, Section VII concludes the paper.

II. Related Work

Winfield [33] presents the basic components of a teleoperated system: the robot (remote vehicle), the remote place of work (teleoperation station) and the connectivity between the two components, while different approaches for a Teleoperated Driving system design [15], [26] already exist. Chucholowski *et al.* [9] measured the latency of video-streams over 3G networks while driving. Their measurements reveal a highly varying average latency of 121 ms and state that 3G connections may be sufficient for Teleoperated Driving. Kang *et al.* [20] measured latency transmitting a video stream over LTE and experienced 100 ms of delay. Keon Jang *et al.* [19] investigated the throughput of 3G and 3.5G while driving with cars and high speed trains. A considerable difference between stationary and mobile measurements was observed, whereby lower throughput over UDP and TCP, higher jitter and packet were witnessed in mobility scenarios when compared stationary conditions. Xiao *et al.* [34] measured the performance of cellular networks by conducting measurements at more than 300 km/h of speed and compared the results to stationary and mobility measurements at lower speeds of 100 km/h. They drove 120 km with vehicles and nearly 5000 km with a high speed train utilizing `iperf` and `traceroute` to measure throughput and latency via smartphones. Lauridsen *et al.* [21] drove about 19,000 km in rural, suburban and urban environments. Radio network scanners and smartphones were utilized to study latency, handover execution time, and coverage of four operational LTE networks. They witnessed LTE coverage of about 99% and an average handover latency of about 40 ms. Li *et al.* [22] compared CUBIC and BBR TCP congestion control while driving on a highway. They measured latency using ICMP and by measuring TCP connect times. In addition, throughput was measured by downloading a file. All measurements were conducted using a smartphone. They observed that latency is predominant in the range of 40 ms to 80 ms. TCP throughput in downloading a file is at a median of about 11 MBit/s. Parichehreh *et al.* [27] conducted measurements to compare three different congestion control algorithms in the LTE uplink. They show that the intended behavior of BBR can be seen, but device packet losses have been observed. Merz *et al.* [25] show that the performance of LTE stays robust up to 200 km/h, identifying the signal-to-noise ratio as an important factor to ensure robustness. While a lot of measurements have been conducted already, it is hard to map these results to Teleoperated Driving specifically. For instance, some studies [9], [19] focus on 3G networks only, some studies lack either throughput [21] or upload measurements[22], [25], while others include limited amount of driving (120 km) [34] or stick to specific routes [27]. Teleoperated Driving is presented in [17], but real measured values are missing. Perhaps results of previous work can still be compared to our study.

Approaches to overcome the issues of mobile networks in Teleoperated Driving have also been proposed. For instance, predictive displays [10], [12] can be used to show the path of the vehicle with respect to the latency and their use can effectively assist while driving. Buffers can be used to overcome the challenges with jitter [14] by smoothening the variability in delay to improve driving performance [12], [23]. An additional possible mitigation, where the remote vehicle automatically reacts to upcoming hazards, which the driver is not yet aware of due to the time delay, is presented in [18]. Another suitable approach is the use of a free corridor [30], where the driver decides the path taken by the car in situations where the connection is lost. In case of uplink throughput, an adaption of resolution is the first step. If conditions occur where such a mitigation is not enough or the resolution becomes too low, the rear camera can be lowered in resolution or even switched off, if not needed. Finally, it is also possible to lower the maximum speed of the remote vehicle and in parallel lower the frames per second of the video stream. For higher speeds it is important to see a fluent image, which is 25fps [29], but with lower speed the frames per second can be reduced and steering is still possible. Finally, it is also possible to use multiple sim-cards [17] of different providers. This enables the option to always use the best network available and transmit important information using multiple paths.

III. Requirements for Teleoperated Driving

We frame the network requirements for Teleoperated Driving within the scope of which Teleoperated Driving may deemed to be feasible with contemporary mobile networks. These requirements consist of minimum/maximum values, that may safely allow Teleoperated Driving. For sending control commands to the remote vehicle, a low amount of data is required. By continuously sending packets of steering commands to the remote vehicle every 10 ms, about 0.25 MBit/s are required. This value is based on the amount of data per packet and the transmission frequency. We decided to send a command packet every 10 ms to get a fluent control stream of steering commands. This allows even small adjustments of the remote vehicle. In addition, with 10 ms between two packets, packet loss/delay can be carried more efficiently as a single packet only counts for 10 ms of driving. The average amount of data per packet originates in measurements, where some basic information were transmitted, e.g. steering wheel angle, position of brake/gas pedals as well as additional triggers like enabled/disabled windshield wiper, etc. Tighter requirements exist for the uplink, which is used at least for streaming video data. For instance, it is known that for transmitting a view of 150° at least 3 MBit/s of uplink [9] are required. This 150° view is deemed sufficient to safely control vehicles in straight driving scenarios, but does not meet governmental regulations, e.g. in Germany. Utilizing a resolution of 640 x 480 and three 90° cameras (front: two, back: one), the amount of transmitted data can be kept at the level of 3 MBit/s [15]. Following the documentation of Youtube for its live-encoder settings [35] and Adobe's recommendations for live streaming [6], 1 MBit/s is deemed enough to carry one stream with sufficient resolution. Thus, sharing three camera streams adds up to about 3 MBit/s. To define the requirements for latency, we further conducted a small user study with five users. The study consisted of driving through `pylons` with different levels of latency, using OpenROUTS3D [3], a self-developed 3D driving simulator. It turned out that values above 300 ms make controlled driving nearly impossible. Subtracting the latency of sensors and actuators (roughly 50 ms), the maximum tolerable network latency is 250 ms. This value correlates with values determined by others [13] for gaming.

We also identified that, if the jitter stays below 150 ms, it is possible to safely control a car.

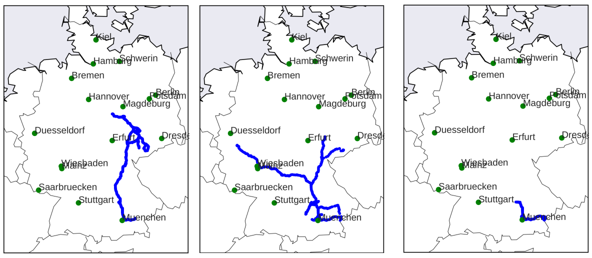
IV. Measurement Setup

We describe the hardware and software setup used to perform measurements of throughput (TCP) and latency (ICMP and UDP) while driving.

Hardware – An Android-based *Lenovo B* smartphone [1] and a *SierraWireless RV50X* LTE gateway [28], installed in a test vehicle, were used to conduct the measurements. The Android-based smartphone was used, to be more flexible as measurements could be carried out independently of the test vehicle. For conducting meaningful measurements, the smartphone needed to allow connectivity to all important mobile network technologies, which the *Lenovo B* does. The LTE gateway on the other hand was fixed to the trunk of the test vehicle. There it was connected to two antennas. One on the roof of the car, which provided LTE and GPS signals, and a second antenna inside the vehicle's trunk to provide diversity and lower the impact of interference. The Gateway was connected to an Ubuntu-based car PC via Ethernet.

Software – Three different tools, `ping`, `netradar` and `iperf3` were used. Tools were not executed in parallel to avoid side effects of reduced throughput or higher latency. The first measurements conducted for this paper consist of ICMP messages sent using a self-developed application [4] for Android-based smartphones. This application is able to gather environmental data for the measurements and execute the `ping` command to determine the Round Trip Time (RTT). While the application allows periodic (configurable) measurements, the configuration applied for this paper had no pause between measurements. Further measurements on the smartphone were collected using the `netradar` [8] measurement platform. `Netradar` is a crowd-sourced mobile measurement platform that measures and collects metrics related to mobile network performance across mobile devices. The measurement mainly focuses on the analysis of TCP throughput [32], UDP latency and contextual information related to each measurement. The server to which `netradar` connects is hosted at the Amazon Cloud in Europe. The throughput-measurements (uplink and downlink) using `iperf3` were conducted on the car PC, where the *SierraWireless RV50X* was configured as LTE gateway. The endpoint for measurements was a server hosted in Munich. Measurements ran continuously and gathered contextual information.

Dataset – Data collection considered only values gathered during driving with a car on all types of streets and areas (rural, suburban, urban; evenly distributed at its best) in Germany to avoid the influence of roaming implications [24]. This ensures to be as close to real world Teleoperated Driving scenarios as possible. The measurements cover the period of end of May 2017 to end of December 2017. The total driving time of ∼78 hours accompanies with ∼5200 km of driving. Measurements are split up into 2180 km (1528 minutes) for `ping`, 2670 km (2940 minutes) for `netradar` and 354 km (191 minutes) for *SierraWireless*. Most of the driving was conducted during daytime. The different routes can be seen in Fig. 1. The average driving speed is a little bit higher (∼66 km/h) than the average speed (∼42 km/h) reported [5] by the Germany Automobile Association (ADAC). That is caused by a higher amount of kilometers on the highway compared to an average driving scenario. Although minor deviations exist, the values are comparable. Discrepancy between the high number of kilometers measuring `ping`/`netradar` and the *SierraWireless* can be explained by the easiness of using the different measurement platforms. The measurements with the *SierraWireless* were only possible when driving with the test vehicle, where availability limits the applicability. In contrast, the smartphone could be carried within any vehicle. The measurements were conducted using Vodafone DE as telecom provider offering unlimited traffic with limitations of 100 MBit/s in downlink and 50 MBit/s in uplink.



(a) Ping (b) Netradar (c) SierraWireless

Fig. 1. Trajectory of (a) `ping`, (b) `netradar` and (c) `iperf3` measurements performed while driving in Germany.

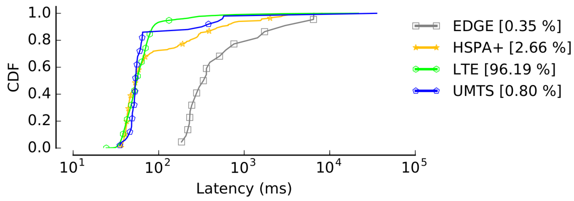


Fig. 2. CDF of the `ping` latencies with overall median of ∼55.14 ms in RTT (EDGE: ∼364 ms, HSPA+: ∼53 ms, LTE: ∼55 ms, UMTS: ∼54 ms).

V. Results

A. Latency

Inflated latency can cause delivery of steering commands and video streams to be delayed. We present results of latencies measured using both ICMP and UDP, since ICMP packets may be treated differently [16] than UDP packets on the path towards the destination.

`ping` – The ICMP packets of the *ping* application were sent either to a server hosted in Frankfurt or Munich. The server hosted in Munich was the target in about 60% of the measurements. The other 40% of the measurements used the server hosted in Frankfurt as target destination.

The connection type of the samples was about 96% LTE, about 3% HSPA+, about 0.8% UMTS and 0.35% EDGE. Mapping this to kilometers means that LTE was used in about 2090 km. UMTS seems to provide results with lower latency

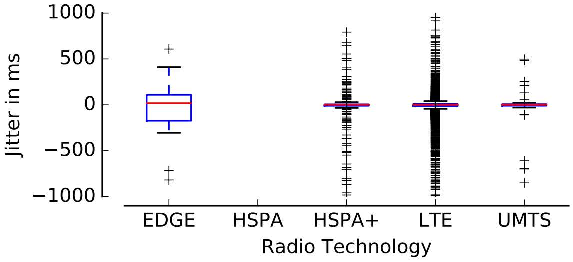


Fig. 3. Variance of `ping` latencies with overall median of ∼10 ms in Jitter.

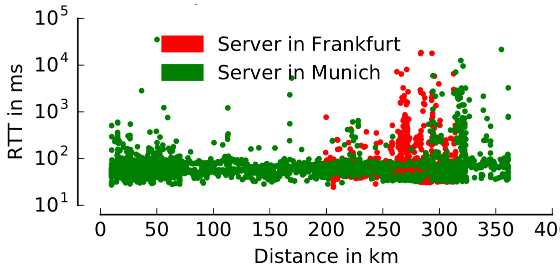


Fig. 5. Distribution of `ping` latency based on destination: Munich/Frankfurt.

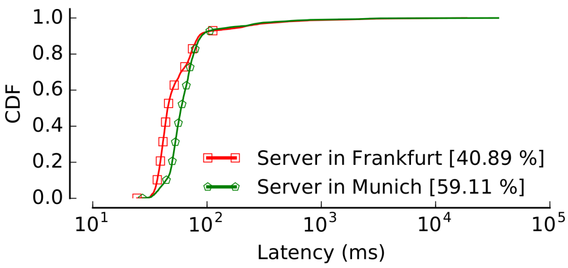


Fig. 4. CDF of `ping` latency based on the destination server in Munich/Frankfurt (Frankfurt: ∼45.3 ms, Munich: ∼59.4 ms in median).

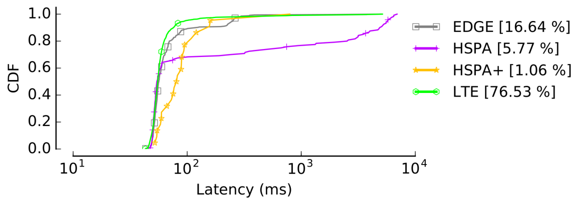


Fig. 6. CDF of the `netradar` latency measurements with overall median of ∼55 ms in RTT (EDGE: ∼56.5, HSPA: ∼54.4, LTE: ∼55.1, HSPAP: ∼81.3).

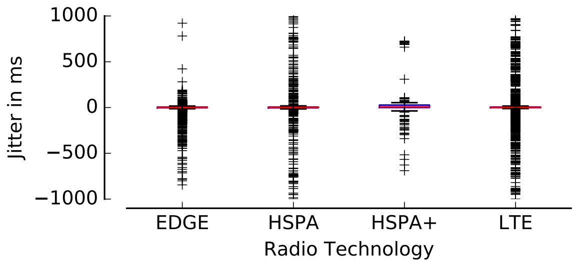


Fig. 7. Jitter of the `netradar` latency with overall median of ∼2 ms.

than LTE, but this is caused by the low number of samples of UMTS measurements (0.8% UMTS, 96% LTE). In addition to the high LTE coverage, the results of the latency measurements are promising as well.

The median latency is ∼55.14 ms in RTT (Fig. 2). About 96% of the total RTT values are below the critical 250 ms. Unfortunately, there are about 4% of higher RTT values, partly greater than one second (1%). This makes Teleoperated Driving infeasible. Besides the raw latency values, jitter (Fig. 3) has to be considered as well. Jitter here references the variance of latency around the median values. For presentation, the jitter has been cut to 1000 ms, as about 99% of the values are below this threshold. The median jitter is at acceptable ∼10 ms (all technologies), but in about 5%, jitter is not suitable for Teleoperated Driving. Jitter comparison between UMTS and LTE faces the same issue as mentioned above.

Given that there may be multiple teleoperation stations deployed at different places, investigating whether the location of a teleoperation station influences the latency is crucial. We chose two servers in cities with about 300 km of distance between each other. Latency to both servers is roughly comparable, but the one to the server in Frankfurt is lower (∼45 ms to ∼59 ms) as shown in Fig. 4. The average distance of the vehicle to the server in Munich and Frankfurt is 119 km and 270 km, respectively. Fig. 5 shows that Munich has greater maximum distances than Frankfurt. Comparing the measurements based on the distance of the vehicle to the server, it can be seen that a higher distance between vehicle and teleoperation station does not significantly lead to higher latency, but its variation drastically increases. As both servers are connected to the Internet with identical parameters, this shows that the location of the teleoperation station is crucial.

`netradar` – The distribution of radio technology types with the `netradar` measurements was about 77% LTE, 17% EDGE, 6% HSPA and 1% HSPA+. Compared to the `ping` application, the LTE coverage was lower in the magnitude of 20%, leading to about 2056 km with LTE as connection type. Besides the differences in the coverage, the median latency, which is ∼55 ms in RTT (see Fig. 6), is comparable to the `ping` results. In about 96% of the samples, the latency was below the critical value of 250 ms. Although the LTE coverage was lower, the results are comparable to the `ping` measurements. The median jitter (Fig. 7, excluding values above 1000 ms (0.2%)) was ∼2 ms and thus more stable than on the `ping` measurements. About 4% of the measurements have a jitter greater than 150 ms.

B. Throughput

The quality of transmitted video streams has to be adapted to the available throughput dynamically. Conducting Teleoperated Driving is infeasible if throughput is too low or changes too frequent for algorithms to adapt.

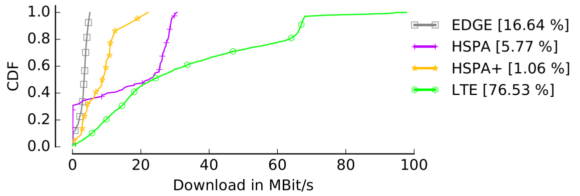


Fig. 8. CDF of `netradar` downlink measurements with overall median of ∼17 MBit/s (EDGE: ∼3.4 MBit/s, HSPA: ∼22.5 MBit/s, LTE: ∼23.5 MBit/s, ∼HSPAP, ∼9 MBit/s).

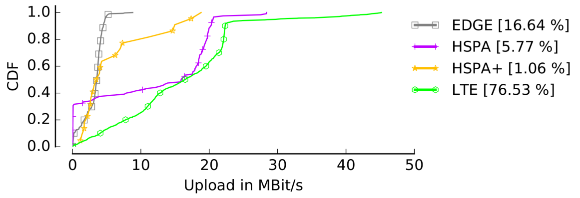


Fig. 10. CDF of `netradar` uplink measurements with overall median of ∼12 MBit/s (EDGE: ∼3.5 MBit/s, HSPA: ∼15.9 MBit/s, LTE: ∼16.5 MBit/s, HSPAP: ∼3.7 MBit/s).

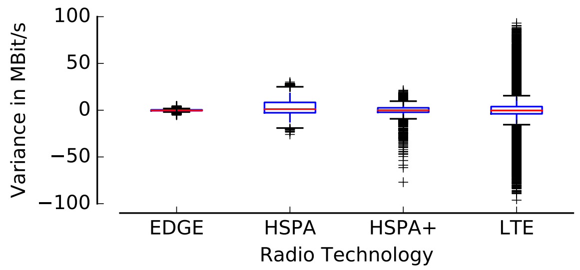


Fig. 9. Variance of `netradar` downlink measurements using different technologies with the overall median of ∼0.15 MBit/s.

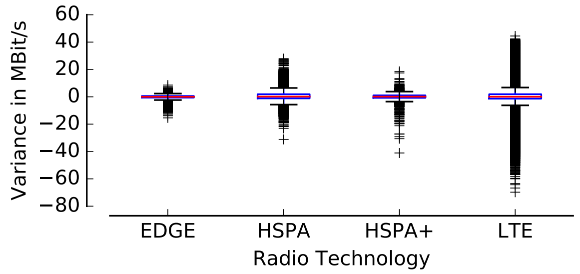


Fig. 11. Variance of `netradar` uplink with overall median of ∼0.07 MBit/s.

`netradar` – Fig. 8 shows the CDF of downlink measurements. As can be seen, the downlink of HSPA and HSPA+ seem to behave unexpected; HSPA seems to be faster than HSPA+. This is caused by the low number of measurements with those network technologies. The median downlink is ∼17 MBit/s. In about 95% of the measurements, the downlink speed is above 0.25 MBit/s and thus sufficient for Teleoperated Driving. The median variance is acceptable ∼0.15 MBit/s, but the maximum variance of ∼96 MBit/s is very high (Fig. 9).

Adequate uplink speed is required to provide qualitative sufficient video streams. The median uplink speed is ∼12 MBit/s (Fig. 10). In about 87% of the measurements the uplink speed is above 3 MBit/s and thus sufficient for Teleoperated Driving. While the median variance is acceptable ∼0.07 MBit/s, the maximum variance is ∼70 MBit/s and thus very high (Fig. 11). In general, these throughput results seem viable for Teleoperated Driving, but there is a high percentage of about 13%, where the uplink speed is insufficient. If only considering LTE connectivity, uplink and downlink is sufficient in about 95% of the time.

SierraWireless – The setup was connected to LTE in about 91.3% and to UMTS in 8.7% of the time. Compared to the `netradar` measurements, the connectivity is better, but with less samples. The median downlink speed is ∼28 MBit/s (Fig. 12). This is nearly double the value of the `netradar` results. Downlink is sufficient for Teleoperated Driving in more than 99% of the samples. The median variance in the downlink is ∼0.41 MBit/s, with a maximum variance of ∼43 MBit/s. The median uplink speed (Fig. 13) is ∼18 MBit/s, which is higher compared to the `netradar` results. The uplink speed is sufficient for Teleoperated Driving in about 98% of the measurements. The median variance in the uplink is ∼0.07 MBit/s, with a maximum variance of ∼26 MBit/s.

C. Measurements on Identical Routes

A comparison of `netradar` and `ping` measurements, where both were on the identical route were further examined. The median latency for `netradar` on this route is ∼55 ms, whereas `ping` shows ∼57 ms in RTT. The connectivity was about 97% LTE and 3% UMTS at the `ping` measurements and about 96% LTE and 4% EDGE during the `netradar` measurements. Results are roughly comparable with the same Hardware and different measurements techniques.

We further study whether the measurement platform makes a significant difference. The median downlink speed of `netradar` on the specific route is ∼15 MBit/s, whereas the downlink of *SierraWireless* is ∼32 MBit/s. The median uplink speed of the `netradar` measurements is ∼13 MBit/s, at the *SierraWireless* it is ∼20 MBit/s. Even with these measurements, there is a significant difference between both systems. The connectivity of `netradar` was about 91% LTE and 9% EDGE, whereas the connectivity of *SierraWireless* was 100% LTE on the identical route. Only considering the LTE connectivity of `netradar`, the uplink is ∼14 MBit/s and the downlink is ∼16 MBit/s. That is not a significant difference compared to the overall measurements on this route. Thus, the *SierraWireless* results are still better. This most likely is attributed to the setup with two antennas. Compared to the antenna inside a smartphone, they seem to provide better results. Thus, the use of better hardware, e.g. more antennas and better positioning of them, can improve feasibility of Teleoperated Driving.

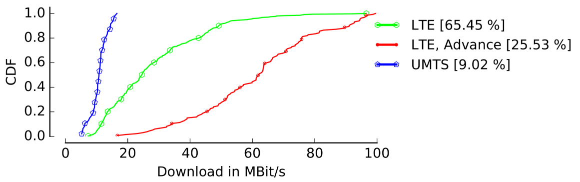


Fig. 12. CDF of *SierraWireless* downlink with overall median of ∼28 MBit/s (LTE: ∼24.5 MBit/s, LTE-A: ∼62.1 MBit/s, UMTS: ∼10.8 MBit/s).

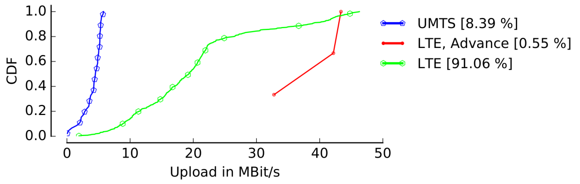


Fig. 13. CDF of *SierraWireless* uplink with overall median of ∼18 MBit/s (LTE: ∼19.2 MBit/s, LTE-A: ∼42.2 MBit/s, UMTS: ∼4.5 MBit/s).

D. Comparison of Different Scenarios

In addition to the comparison of different measurement setups at the identical routes, factors such as vehicle speed, handover, signal-strength, distance to the base station and their influence on the performance were further investigated.

Handover – We wanted to investigate whether handover has an influence on Teleoperated Driving. Both types of handover were investigated, e.g. switch with and without changing radio technology. The `netradar` measurements consist of only 12 switches in cell and 19 switches of radio technology. With *SierraWireless*, we witnessed 60 cell switches and no radio technology changes during the downlink measurements with 54 and one switch respectively with uplink measurements.

The overall median latency of the `netradar` measurements is ∼55 ms. When only switching cells, but keeping the same radio technology, there is no negative influence on the latency. When switching network technology during a measurement, latency increases by about 15%. In case of throughput, the median downlink is ∼17 MBit/s. With a cell-only switch, this value decreases to ∼14 MBit/s. In case of switching radio technology, the downlink drastically decreases to ∼3 MBit/s. The median uplink speed is ∼12 MBit/s. Keeping the same radio technology does not influence this value, but when changing the network technology, the uplink speed decreases to ∼3 MBit/s.

The median downlink speed for *SierraWireless* is ∼28 MBit/s. When performing a handover by keeping the same radio technology, the speed does not change. The median uplink speed is ∼18 MBit/s. When changing only the cell but keeping the radio technology, the uplink speed stays the same. In case of changes in the network technology, there is also no real difference, but the number of switches is only one. In general it can be said, that Teleoperated Driving is feasible when switching cells but keeping radio technology and infeasible when switching radio technology.

Speed – With Teleoperated Driving, the remote vehicle will have different speeds based on where it is driven. Considering the speed, we wanted to investigate whether different levels of speed have influence on the latency or throughput. We observe, for Teleoperated Driving, there is no real influence of the speed of the vehicle regarding latency or throughput. Even if there is a slightly higher performance at higher speeds with the *SierraWireless* measurements, this observation is only because speeds above 100 km/h were conducted on the highway, which provide a better network coverage with base-stations similar to observations made by previous studies [25].

Signal Strength – Due to the mobility of the remote vehicle, it can happen that the signal strength changes, e.g. in tunnels, and thus might have negative influence on Teleoperated Driving. In case of throughput (uplink/downlink), there is a clear tendency. The better the signal strength, the higher the throughput. For latency, no clear tendency is witnessed.

Distance – Finally, a vehicle when driving will change its distance to a base-station. We analyzed this distance based on the data from OpenCellID [2] to investigate whether there is an influence either on latency or the throughput. The distance to the base station was usually less than 5 km and no obvious influence of the distance on the values for throughput (uplink/downlink) and latency were observed.

E. Whitelisting as Possible Mitigation

In general, latency, jitter and throughput values are promising for Teleoperated Driving. However, our measurements are limited and only reflect the network states of the routes taken. We believe this to be acceptable for a first assessment, but not exhaustive for real world use cases. Therefore, we propose the approach of whitelisting with frequent probing. This whitelisting is a simple approach to mitigate critical situations by allowing Teleoperated Driving only in areas that are measured to provide sufficient network performance. To examine, whether Teleoperated Driving would be feasible in areas with good network connectivity, a typical whitelisting scenario, we drove in an area with LTE or LTE-Advanced coverage only. We chose a time in the afternoon at which a lot of commuters are driving, to get measurements with a high number of other road users. The driven route is a 5 km long circle around the historic center of Ingolstadt. We drove on it four times, leading to a total of about 20 km (60 minutes). The driving activity was split up into two parts. The first part consisted of two rounds for measuring over TCP. The second part consisted of two rounds for measuring over UDP, resulting in about 10 km and ∼30 minutes for each part. The standard Linux `ping` utility on the car PC was executed all the time. Both parts were driven consecutively without breaks in between. The measurements were conducted with the test vehicle and the aforementioned *SierraWireless* setup. In contrast to the previous measurements, a maximum bandwidth of 5 MBit/s was specified to be used. Thus, we are able to explore how stable values are. We chose 5 MBit/s to have the minimum required 3 MBit/s for uplink plus additional 2 MBit/s as margin. Measurements on the first part

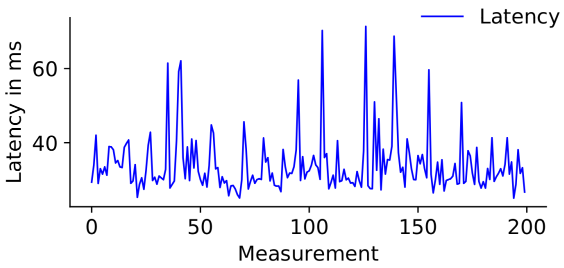
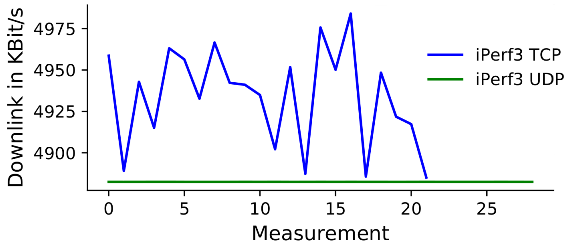


Fig. 14. Latency measured during the test drives with median of ∼31 ms.

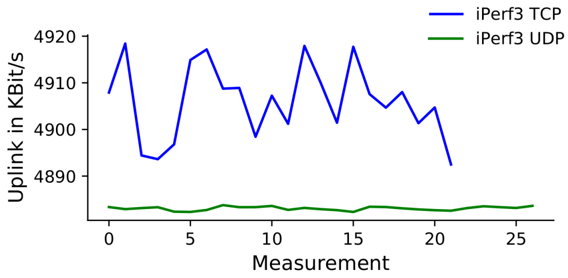
were conducted continuously with the sequence: (1) `ping`, `iperf3` download (TCP), `ping` (2) `ping`, `iperf3` upload (TCP), `ping`. TCP was replaced by UDP in the second part of the activity. Due to the fact that `ping` is measured more frequently, we have a higher number of `ping` measurements than that of TCP (or UDP).

The measured latency (Fig. 14) has a median of ∼31 ms, whereas the maximum is 71 ms. To investigate whether the `ping` values can be used, we also measured the RTT during the TCP measurements. The median value of ∼27 ms is comparable, but the maximum of 45 ms is lower. Hence, the 71 ms sample point seems to be an outlier. During our test drive in the white listed area, there is no measurement in which the latency is above the critical value of 250 ms, which means that Teleoperated Driving is possible. Jitter is always below 150 ms and thus no issue there either.

We also measured the uplink and downlink speed with TCP and UDP. Fig. 15a shows measured downlink over time. The median downlink speed is ∼4.94 MBit/s (TCP) and ∼4.88 MBit/s (UDP), with minimum values of 4.88 MBit/s (TCP) and 4.88 MBit/s (UDP). This is above the minimal required value of 0.25 MBit/s. Moreover, the variance of values is not critical for Teleoperated Driving. Fig. 15b shows the the uplink speed. The median uplink speed is ∼4.90 MBit/s (TCP) and ∼4.88 MBit/s (UDP), whereas the minimum values are 4.89 Mbit/s (TCP) and 4.88 MBit/s (UDP). These values are above the required minimum of 3 MBit/s. The fluctuation of the uplink speed leaves Teleoperated Driving feasible. For upload and download, the values never reach the specified 5 MBit/s, but that is based on the measurement method of `iperf3`. During our measurements 16 handovers without and 5 with network technology changes occurred. However, it can be seen that there is no real influence on the performance. The packet loss during the UDP measurements is in the order of $10^{-4}\%$ and thus less an issue for Teleoperated Driving. In general it can be said that a whitelisting-based approach could work. Our results confirm the applicability of whitelisting [17] for Teleoperated Driving. We are aware that our measurements provide a snapshot of the network state, altough we attempt to select a route and a time with a real amount of traffic. The whitelisting approach has to be advanced further by permanently probing areas, e.g. normal vehicles conducting periodic network performance measurements and by sending



(a) Downlink



(b) Uplink

Fig. 15. Downlink (median ∼4.94 MBit/s for TCP, median ∼4.88 MBit/s for UDP) and uplink (median ∼4.90 MBit/s for TCP, 4.88 MBit/s for UDP) measured during the test drives.

them to a cloud service. The cloud service can be advanced with data, following the predictability of connected cars [7]. Using this cloud-service, teleoperated vehicles may plan and update their route based on the incoming data. If situations are changing dynamically, adjustments can be applied early before a remote vehicle enters a dangerous area. With the increased number of measurements, a more accurate map can be built and Teleoperated Driving might be possible. A complementary blacklisting can also follow, so that areas where Teleoperated Driving will not be possible are blocked. Although using such a navigation-based approach can help, network parameters can become critical even within whitelisted areas. For such situations, other approaches, e.g. based on the vehicles' monitoring system [26] have to be applied.

VI. Limitations and Future Directions

The results presented in this paper are limited by the amount and type of measurements we conducted. Changes in network conditions are always likely to occur and will influence the measurements we have conducted. The results reflect end-user's perspective and do not consider how providers treat different packets. We treat the network connection as black box, e.g we are thus not able to obtain information on how busy particular cells were. The whitelisting approach is only presented in principle and does not yet consider changes in network conditions. Nevertheless, these results can be used to get a first impression whether Teleoperated Driving could be feasible with contemporary mobile networks.

In the future, a user study with more participants will be conducted, to investigate whether skilled and trained drivers are able to cope with network conditions of high delay. The cloud-based navigation system will be improved and tested, to see, whether the approach is feasible in everyday scenarios.

VII. Conclusion

We observed that Teleoperated Driving may be feasible with contemporary mobile networks, especially with LTE and LTE-Advanced capability in whitelisted areas. In most cases (`ping` and `netradar`: 96%), the latency was below 250 ms, the uplink was above 3 MBit/s (*SierraWireless*: 98%, `netradar`: 87%) and the downlink above 0.25 MBit/s (*SierraWireless*: 99%, `netradar`: 95%). This indicates that in the majority of our measurements, Teleoperated Driving could be used. With the test drive on a specific route, we showed that the whitelisting approach can work, but the basic approach has to be improved considering changes in network conditions to provide accurate maps for Teleoperated Driving. Frequent probing and providing results to a cloud-based service might help. Teleoperated vehicles are then able to dynamically adapt their route with changes in network conditions. There are also cases in which Teleoperated Driving does not work. Latencies above 1 second, high jitter and low throughput are not tolerable and have to be avoided. Further, handover between two cells can negatively influence Teleoperated Driving, as it might lower the throughput and increases the latency. The signal strength has influence on the throughput but not on the latency. The better the signal strength, the higher the throughput. The position of the teleoperation station is also crucial. Fluctuation of the latency increases if the remote vehicle is further away from the teloperation station. Improved hardware and multiple antennas can help with the connectivity and thus also increase the usability of Teleoperated Driving. In general it can be said that at least within our measurements, Teleoperated Driving can be feasible with contemporary mobile networks, but more measurements and clever approaches are required.

References

[1] Lenovo B smartphone. https://lnv.gy/2VvFZ50. Accessed: 27.09.2018.

[2] OpenCellID. https://www.opencellid.org. Accessed: 15.05.2019.

[3] OpenROUTS3D. https://git.io/fj8tn. Accessed: 16.05.2019.

[4] ping utility for Android. https://bit.ly/2Hr4Tzi, 2019.

[5] ADAC. Mobility in Germany - Selected Results (org: Mobilitaet in Deutschland - Ausgewaehlte Ergebnisse). https://bit.ly/2w0qRCD, 2010.

[6] Adobe Developer Connection. Recommended bit rates for live streaming. https://adobe.ly/2wd8pH7. Accessed on 11.10.2018.

[7] Carlos E. Andrade, Simon D. Byers, Vijay Gopalakrishnan, Emir Halepovic, David J. Poole, Lien K. Tran, and Christopher T. Volinsky. Connected Cars in Cellular Network: A Measurement Study. In *Internet Measurement Conference*, 2017.

[8] Vaibhav Bajpai and Jürgen Schönwälder. A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts. *IEEE Communications Surveys and Tutorials*, 17(3):1313–1341, 2015.

[9] Frederic Chucholowski, Tito Tang, and Markus Lienkamp. Teleoperated Driving Robust and Secure Data Connections. *ATZelektronik worldwide*, 9(1):42–45, Feb 2014.

[10] Frederic Emanuel Chucholowski. Evaluation of Display Methods for Teleoperation of Road Vehicles. *Journal of Unmanned System Technology*, 3(3):80–85, 2016.

[11] ERTRAC Working Group "Connectivity and Automated Driving". Automated Driving Roadmap. resreport, ERTRAC, July 2017.

[12] J. Davis, C. Smyth, and K. McDowell. The Effects of Time Lag on Driving Performance and a Possible Mitigation. *IEEE Transactions on Robotics*, 26(3):590–593, June 2010.

[13] Matthias Dick, Oliver Wellnitz, and Lars Wolf. Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games. In *ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames '05, pages 1–7, New York, NY, USA, 2005. ACM.

[14] P. M. d'Orey, A. Hosseini, J. Azevedo, F. Diermeyer, M. Ferreira, and M. Lienkamp. Hail-a-Drone: Enabling teleoperated taxi fleets. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 774–781, June 2016.

[15] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. A System Design for Teleoperated Road Vehicles. In *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, pages 231–238, July 2013.

[16] Hang Guo and John Heidemann. Detecting ICMP Rate Limiting in the Internet. In Robert Beverly, Georgios Smaragdakis, and Anja Feldmann, editors, *Passive and Active Measurement*, pages 3–17, Cham, 2018. Springer International Publishing.

[17] Mark Harris. CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads. https://bit.ly/2D26Z87, 2018. Accessed:28.11.2018.

[18] A. Hosseini and M. Lienkamp. Predictive safety based on track-before-detect for teleoperated driving through communication time delay. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016.

[19] Keon Jang, Mongnam Han, Soohyun Cho, Hyung-Keun Ryu, Jaehwa Lee, Yeongseok Lee, and Sue B. Moon. 3G and 3.5G Wireless Network Performance Measured from Moving Cars and High-speed Trains. In *ACM Workshop on Mobile Internet Through Cellular Networks*, 2009.

[20] Lei Kang, Wei Zhao, Bozhao Qi, and Suman Banerjee. Augmenting self-driving with remote control: Challenges and directions. In *Workshop on Mobile Computing Systems & Applications*, 2018.

[21] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sorensen, and P. Mogensen. From LTE to 5G for Connected Mobility. *IEEE Communications Magazine*, 55(3):156–162, March 2017.

[22] Feng Li, Jae Won Chung, Xiaoxiao Jiang, and Mark Claypool. TCP CUBIC versus BBR on the Highway. In *Passive and Active Measurement Conference*. Springer International Publishing, 2018.

[23] Ruilin Liu, Daehan Kwak, Srinivas Devarakonda, Kostas Bekris, and Liviu Iftode. Investigating Remote Driving over the LTE Network. In *Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '17, 2017.

[24] Anna Maria Mandalari, Andra Lutu, Ana Custura, Ali Safari Khatouni, Özgü Alay, Marcelo Bagnulo, Vaibhav Bajpai, Anna Brunström, Jörg Ott, Marco Mellia, and Gorry Fairhurst. Experience: Implications of roaming in europe. In *MobiCom 2018*, pages 179–189, 2018.

[25] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. Performance of LTE in a High-velocity Environment: A Measurement Study. In *Workshop on All Things Cellular: Operations, Applications, & Challenges*, AllThingsCellular '14, 2014.

[26] Stefan Neumeier, Nicolas Gay, Clemens Dannheim, and Christian Facchi. On the Way to Autonomous Vehicles - Teleoperated Driving. In *AmE 2018 - Automotive meets Electronics*, 2018.

[27] A. Parichehreh, S. Alfredsson, and A. Brunstrom. Measurement Analysis of TCP Congestion Control Algorithms in LTE Uplink. In *Traffic Measurement and Analysis Conference*, pages 1–8, June 2018.

[28] SierraWireless. AirLink RV50 Datasheet. https://bit.ly/2W7dkrj. Accessed on 27.09.2018.

[29] Tito Tang, Frederic Chucholowski, and Markus Lienkamp. Teleoperated driving basics and system design. *ATZ worldwide*, 116(2), Feb 2014.

[30] Tito Tang, Pascal Vetter, Markus Lienkamp, Simon Finkl, and Korbinian Figel. Teleoperated Road Vehicles – The "Free Corridor" as a Safety Strategy Approach. In *Mechanical Design and Power Engineering*. Trans Tech Publications, 2014.

[31] Ermias Andargie Walelgne, Setälä Kim, Vaibhav Bajpai, Stefan Neumeier, Jukka Manner, and Jörg Ott. Factors affecting performance of web flows in cellular networks. In *IFIP Networking Conference*, 2018.

[32] Ermias Andargie Walelgne, Jukka Manner, Vaibhav Bajpai, and Jörg Ott. Analyzing Throughput and Stability in Cellular Networks. In *IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2018.

[33] Alan Ft Winfield. Future Directions in Tele-operated Robotics. In *Telerobotic Applications*. 2000.

[34] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong. TCP Performance over Mobile Networks in High-Speed Mobility Scenarios. In *Conference on Network Protocols*, pages 281–286, Oct 2014.

[35] YouTube Help. Live encoder settings, bitrates and resolutions. https://bit.ly/2oob5w0. Accessed: 11.10.2018.

Publication [III]: Towards a Driver Support System for Teleoperated Driving

[**III**] Stefan Neumeier and Christian Facchi. “Towards a Driver Support System for Teleoperated Driving”. In: *22nd Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct. 2019

The main idea behind this paper is that adjusting the speed of the remote vehicle according to latency, allows for a safe control of the vehicle. Therefore, we had to take three different factors into account: The distance to the vehicle ahead, the maximum allowed speed based on the stopping distance and the ability to drive safely through curves. In case of distance to the vehicle ahead, a greater distance to the vehicle ahead can counteract the longer distance the vehicle requires in case of an emergency brake. Therefore, the distance to the vehicle ahead needed to be enlarged by considering the driving speed and the latency of system and network. E. g., instead of keeping a distance equivalent to half-of-tachometer it would be half-of-tachometer plus an additional latency induced distance. Another point was the reduction of the speed in order to keep the same stopping distance as if there was no latency, i. e., doing emergency brakes for unexpected obstacles. This can easily be achieved by reducing the vehicle's speed in a magnitude that still allows to stop as without latency, e. g., if driving 30 km/h one would drive for example 28 km/h to counteract the latency by stopping at the same total distance. Finally, it was crucial to address the driving through curves, as steering commands are executed delayed. This can lead to a dangerous situation, as the curve is entered with a speed above the one which is considered for a safe and comfortable driving. Therefore, the speed had to be reduced accordingly to the latency. Putting everything together was possible – up to a certain level – to counteract packet loss and address compression and pre-processing times, e. g., as for bandwidth reductions. It turned out that the speed adjustments are not that strong as one would expect, e. g., the speed only had to be reduced only by about 3 km/h on the described scenario, using real-world latency values.

The initial idea to this paper is based on thoughts I had on counteracting latency in teleoperated driving by adjusting the actual driving speed. In close exchange with the co-author I developed the presented approaches and did their final assessment based on the first results on the real-world measurements. The major part of the writing was done by me and happened in close exchange with the co-author. I also mainly took care of addressing reviewers' comments.

IEEE Requesting permission to reuse content from an IEEE publication

Towards a Driver Support System for Teleoperated Driving

Conference Proceedings: 2019 IEEE Intelligent Transportation Systems Conference (ITSC)

Author: [::Stefan::] [::Neumeier::]; Christian Facchi

Publisher: IEEE

Date: 27-30 Oct. 2019

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Towards a Driver Support System for Teleoperated Driving

Stefan Neumeier[1] and Christian Facchi[1]

Abstract— **Numerous problems regarding autonomous driving were solved in recent years. However, there are still limitations preventing the introduction of autonomous driving features beyond SAE level 3. A remote operator can be a promising fallback solution. Teleoperated Driving is the remote control of vehicles by a human operator located at an operation center. Existent infrastructure such as cellular networks has to be utilized to provide a functional solution. However, cellular networks suffer from variable bandwidth, variable delay and packet loss. Thus, a route planning based on previous network measurements is proposed. As network parameters vary over time, speed adjustments are suggested to allow the remote operator to react quickly enough in any way. It is shown that the approach can work in principle, but further long-term measurements are required to identify and eliminate potential issues.**

I. Introduction

Development in recent years solved numerous problems regarding autonomous driving, but there are still many limitations that prevent the deployment of autonomous driving beyond SAE [1] level 3. This limitations might be complex roadside works, e.g. as in [2], or other obstacles as shown by Kang et al. [3]. A fallback solution to this might be a remote operator. Teleoperated Driving means to hand-over the control of a vehicle to a human operator located in dedicated operation centers. Such systems are currently developed, partly already in use and required by law, e.g. in California for driverless vehicles [4]. Applying such a technology in a functional way requires the use of existent infrastructure such as cellular networks, which must be capable for Teleoperated Driving. Modern technologies like Long Term Evolution (LTE), LTE-Advanced [5] and 5G [6] are sufficient for Teleoperated Driving. Unfortunately, theory and practice differ. Cellular networks typically suffer from variable and high latency, variable and low bandwidth, packet loss and no coverage at all. Thus, a route planning system that only considers areas with sufficient network performance is proposed. However, network performance varies over time and an isolated route planning system is not sufficient. Therefore, a second step consisting of speed adjustments is proposed. Speed adjustments will happen based on the network performance and allow a remote operator to react in time. Thus, this paper answers the question if Teleoperated Driving could be possible by applying the proposed approach. However, this does not solve all issues, but first results promise that route planning and adjustment of speed in principle can be used to safely operate remote vehicles. To do so, some simplifications, mainly in the mathematical descriptions, are applied. This does not change the validity of the results, but allows the introduction of the approach without considering varying parameters like vehicle weight, vehicle type, wind, etc. The paper is structured as follows. Section II gives an overview of related work. Section III introduces the twofold approach. Section IV presents some basic results. Section V draws a conclusion and suggests future work.

[1] Research Centre, Technische Hochschule Ingolstadt {stefan.neumeier, christian.facchi}@thi.de

II. Related Work

Teleoperated systems typically consist of three parts: Teleoperated device (robot), teleoperation workspace and communication link [7]. The teleoperated device is the remote vehicle. Usually it is equipped with sensors that allow a remote operator (teleoperator) to get an idea of the environment. Typically, these are cameras, but other sensors are conceivable. An interface, displaying sensor data and providing controlling, is available for the teleoperator. Finally, a connection between teleoperator and remote vehicle is required to exchange steering commands and sensor data. Basic research regarding teleoperated vehicles has already been carried out, e.g. by Tang et al. in [8] and [9]. A survey on teleoperation, covering aspects like telepresence and control issues, was done by Lichiardopol [10]. Teleoperation setups are shown for instance by Neumeier et al. [11], Gnatzig et al. [12] and Shen et al. [13]. In general, research focuses on usage studies for teleoperated vehicles, visualization of data at the teleoperator cockpit and impacts of time delay and data rate. For example, Chucholowski et al. [14] report time delays in the range of 65 ms to 1299 ms if transmitting image data via 3G. Shen et al. experienced a latency ranging from 143 ms (min. 4G) to 463 ms (max. 3G) transmitting a video stream [13]. 100 ms of latency were observed by Kang et al. [3] transmitting a video stream over LTE. Basically, mobile connections suffer from potentially high delays and packet loss [15]. The data rate, furthermore, can drop drastically depending on the mobile cell workload. 5G could mitigate these problems and provide dedicated communication channels [5], but a roadmap shows first 5G installations by 2020 [16], initially covering only specific areas. In addition to data compression, current approaches employ lightweight protocols like the User Datagram Protocol (UDP) in order to reduce communication overhead [14]. Variable latency of wireless connections makes it hard to safely control the vehicle [17]. Various research demonstrated several approaches to allay this so-called time-lag problem. In [17] Davis et al. have shown, that the use of a predictive display can mitigate the impacts of lags by indicating the latency based state, e.g. foreshadowing the time delay based car position. In [18] Chucholowski compared various types of predictive displays in a study, showing that their usage can effectively assist the driver with his task. In [19] Hosseini and Lienkamp present an approach in which the vehicle reacts automatically to yet unknown upcoming hazards, for instance by adjusting the speed. A different suitable approach shown by Tang et al. is the use of a free corridor, where the driver has to set the vehicle's trajectory in a way, that it still can move safely even if the connection is lost [20]. Situation awareness is crucial for Teleoperated Driving. This situation awareness can be achieved best if the teleoperator is aware of its relevant environment [21] and position [22]. In [23] Hosseini and Lienkamp show that utilizing virtual reality glasses and combining available sensor data to a representation of the environment can help to improve situation awareness. This is achieved by merging camera-based video with 3D models built from LiDAR data. In [24] Georg et al. conducted an user study with virtual reality head-mounted displays and conventional monitors. Their results show, that for slow-speed it does not make any difference.

This paper proposes a novel approach. It focuses on route planning and speed adjustments to allow safe Teleoperated Driving in which operators can react in time.

III. Methodology

To safely enable Teleoperated Driving a twofold approach is proposed. It consists of route planning combined with speed adjustments while driving.

A. Network Coverage

The proposed approach is based on the concept of whitelisting and blacklisting areas where Teleoperated Driving might be possible, e.g. as mentioned in [2]. Areas, as a collection of street sections, that could support Teleoperated Driving are selected with respect to the network performance. Measuring only once is not sufficient as network conditions vary, so a time-considering model is required. This requires a cloud-based backend where live data is gathered and made available for teleoperated vehicles. Based on previous research [25], lower limits of 3 MBit/s in uplink, 0.25 MBit/s in downlink and an upper limit of 300 ms in latency are considered. These values may change in future.

1) Initial map creation: The first step of map creation is locating base stations, e.g. with a service like OpenCelliD [26]. Based on the locations, regions with certainly no network coverage are identified and immediately blacklisted.

2) Probing specific areas: Real-world measurements are the subsequent step. They have to be conducted frequently while driving, as network performance varies. It is imaginable that future vehicles carry hardware and software that can conduct measurements periodically and share their results. Measurements have to be classified appropriately to take differences in the network load into account. Route planning is based on these measurements and only considers areas where network performance is assumed sufficient during

Fig. 1. Overview of blocked (red) and allowed (green) streets/parts of streets based on measured points. The blue marker indicates a transition between blocked and allowed (from [27]).

driving. The route changes dynamically if arriving measurements during driving reveal that network performance on the route changed drastically disallowing safe Teleoperated Driving. Such a system requires the prediction of network parameters, which is a tricky task. Within this paper, a basic approach is described. In a first step, measurements are simply classified by usual and unusual days. Usual days are workdays, whereas unusual days consist of Sundays and bank holidays. Both are tracked based on hourly time-slices. This allows a distinction of how much traffic is on the street and how this is reflected in the network performance. For route planning this means, that if a route is planned, the past measurements will be used to make a prediction based on current an previous parameters and decide which streets will be allowed and which will be blocked. As mentioned later in this paper, blocking and unblocking is only the first step of route planning. This approach has to be adjusted in future by using more detailed information such as weather, special events, etc. along with machine learning techniques. Finally, the remote vehicle frequently probes itself to react on network parameters that differ from estimated/retrieved ones. In general, this probing and prediction allows the blocking and allowing of streets or parts of streets. Driving is only allowed on streets that are not blocked. Such a map based on the measurements of Neumeier et al. [25] can be seen in Figure 1. Green dots indicate streets or parts of streets that might be used, whereas red dots indicate a blocked street or parts of a street. It can be seen that it is sometimes required to block only parts of a street. Blocking will always start/end at intersections as for example indicated with the blue marker in Figure 1.

B. Speed Adjustments

In addition to network performance-based route planning, adjusting the vehicle's speed based on latency is crucial. The authors argue that lowering the vehicle's speed, if latency increases, will technologically still allow a safe and controllable remote driving. Human perception is addressed in a future paper and neglected within this approach. To achieve a safe driving with latency, three parameters are taken into account: Distance to vehicles ahead, maximum speed based on stopping distance, maximum speed driving safely through curves.

1) Distance to vehicles ahead: For dealing with sudden braking of vehicles ahead, maintaining a minimal distance to them is important. It should be large enough and yet avoid disrupting normal traffic flow. For Teleoperated Driving the distance to a vehicle ahead should follow the same rules as with driving non-remotely. For Germany this rule requires a minimal safety distance of half-of-tachometer [28]. Even if usually Time-To-Collision (TTC, e.g. as done for Forward Collision Systems [29]) is used, teleoperators might not be able to estimate a TTC that is precise enough [30]. Therefore, the half-of-tachometer rule seems to be more applicable. The formula used by Beisel et al. [28] can be easily extended with latency (Formula 1). The formula considers speeds of the two vehicles (front: v_f, ego: v_e), deceleration (front: a_f, ego: a_e) as well as network t_l, system t_{sy} and reaction t_r delays to calculate the minimal safety-distance in seconds t_s. With telepresence [24] as influencing factor, the authors argue that time differences can be approximated easier than distances in meters.

$$t_s = (t_l + t_{sy}) + t_r - \frac{v_f^2 - \frac{a_f * v_e^2}{a_e}}{2a_f * v_e} \tag{1}$$

In Formula 1 system and network latency are added to the time distance one should keep to the vehicle ahead without latency. The typical time distance without latency is about 1.8 seconds (half-of-tachometer) [28].

2) Maximum speed based on stopping distances: To stop a remote vehicle suffering from latency within the same distance as a non-remote one, the speed of the remote vehicle has to be reduced. Reduced speed leads to shorter stopping distances, which compensates for latency. The ideal stopping distance s considering latency can be approximated by Formula 2, which is based on the work of Spielmann and Reuter [31]. v_e is the current speed of the vehicle, a_e is the deceleration rate and l_s, l_n are the system and network latency, respectively. This formula is simplified and thus does not explicitly consider influencing factors like wind, etc. [32], [33], but does so implicit with an appropriate deceleration rate.

$$s = v_e * (l_s + l_n) + \frac{v_e^2}{2 * a_e} \tag{2}$$

Figure 2 shows distances of influencing factors for the stopping distance considering different levels of speed. The deceleration rate a is set to reasonable braking of 4 m/s^2. Based on values of Shen et al. [13], network and system latency are set to 150 ms and 50 ms, respectively. It can be seen that reaction and stopping are major factors, while system and network latency play a minor role. They cause an additional median distance of 1.11 m and 3.33 m, respectively. To achieve braking with latency that is comparable to non-latency braking, speed reduction has to be applied. The basic formula for braking is enhanced by latency. The allowed speed is calculated by Formula 3, assuming a constant deceleration. Reaction time will stay the same as drivers are still humans - only in front of a monitor. The new speed v_l is based on the system l_s and network l_n latency. The initial speed v_0 is the speed someone would drive without latency and is based on speed limits and situation-based personal assessment.

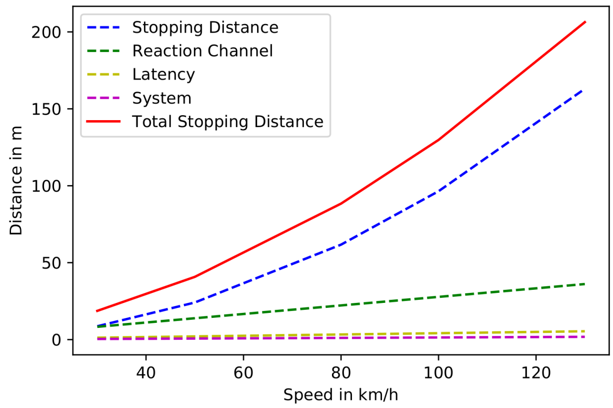


Fig. 2. Covered distances with 150 ms network and 50 ms system latency.

$$v_l = -a * (l_s + l_n) + \sqrt{a^2 * (l_s + l_n)^2 + v_0^2} \tag{3}$$

The new speed v_l is mandatory to ensure that stopping happens within the same distance compared to a scenario without latency. If latency varies, speed has to be adapted frequently. The allowed speed could change drastically if the latency increases, but usually there is no big difference. High latency values disallow Teleoperated Driving and thus can be ignored. In principle the approach of reducing vehicle's speed to overcome latency can work. Even with higher speeds the difference is not too big to become a security risk or obstacle for other traffic participants.

3) Maximum speed to drive safely through curves: Driving through curves is tricky. Vehicle's speed has to be adjusted to the curve's radius. Avoiding uncomfortable driving or sliding off the street, small radii require low speed, whereas large radii allow for higher speed. The authors argue that driving through curves with latency is possible by adjusting the speed. Latency causes steering commands to be executed delayed. This subsequently leads to a higher steering wheel angle and unavoidably to higher lateral acceleration. This will make passengers feel uncomfortable or force the vehicle sliding off the street. A skilled remote driver might anticipate this and steer in advance, but that is very subjective and thus error-prone. Therefore, the proposed algorithm calculates a suitable curve-speed, enabling a safe driving. Lateral acceleration is calculated based on the single-track model [34]. The model is simplified to a steady-state circular [35]. Formula 4, where r is the curve radius, is used to estimate the maximal lateral acceleration based on the typical and uncritical speed during a normal drive.

$$a = v * \dot{\Psi} = \frac{v^2}{r} \tag{4}$$

For this paper, Openstreetmap (OSM) [27] data is used to obtain information about curvature. In OSM a curve is not marked explicitly, but described with single points. Figure

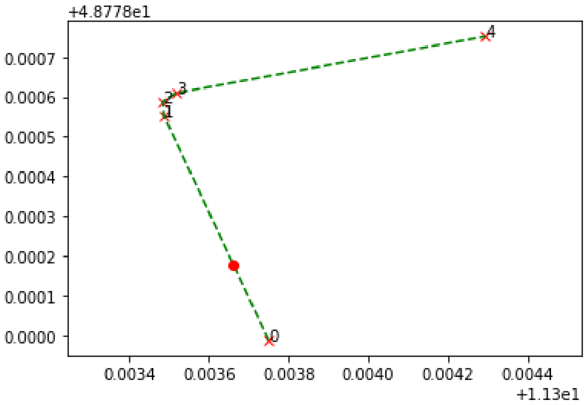


Fig. 3. Markers of the street with the repositioned point for radius calculation.

3 shows numbered markers of OSM, which in combination describe a street and its curvature. The dotted green line indicates the real-world roadway between these points. For the calculation, streets have to be split into different parts consisting of three subsequent markers each. The curvature of a street-part is calculated by placing an outer-circle at 3 distinct points forming a triangle, e.g. points 0,1,2 in Figure 3. For this purpose, the distances in meters between the three points are calculated using the approach of Karney [36]. Factoring in these distances, Formula 5 is used to calculate the radius based on the surface area. The distances between points are indicated by a, b, c, whereas A is the surface area of the triangle.

$$R = \frac{a * b * c}{4 * A} \tag{5}$$

Additionally, Heron's formula (Formula 6) is applied to calculate the up to now unknown surface area [37].

$$A = \frac{\sqrt{(-a+b+c)\,(a-b+c)\,(a+b-c)\,(a+b+c)}}{4} \tag{6}$$

With the calculated radius and the knowledge, that a typical lateral acceleration is at a maximum of about 0.3 m/s^2 in case of a dry street [38], the common curve speed v_c can be calculated by Formula 4. Thus, it can be determined how fast a non-remote vehicle would typically go through the curve under ideal environmental conditions, e.g. no rain, snow, black ice, etc. Considering latency, the speed of a remote vehicle has to be reduced to stay below 0.3 m/s^2 of lateral acceleration. Thus, the latency-based radius is calculated. This can be achieved by repositioning the first point of the curve calculation each time a radius is calculated. Repositioning is based on network latency and calculated speed of the vehicle. Spline-interpolation with first-order splines [39] is applied to estimate the first point's new position. An example of this can be seen in Figure 3, where an exemplary narrow curve was chosen. Usually, the calculation for the first part of the curve consists of the points 0,1,2. Considering the calculated vehicle speed and network latency, the new first point is repositioned (red dot). Therefore, the radius is reduced and a new latency-based speed will be calculated. The previous calculation only considers single parts of a street, but all parts of it are important. For example, Figure 3 is split into three different parts: (0,1,2); (1,2,3); (2,3,4). Thus, the allowed speed of every part of the curve is calculated based on original speed and latency. The subsequent step consists of putting all parts of a street together. The goal of the proposed approach is to avoid braking during cornering. Therefore, continuous curves are identified and the speed through the whole curve will stay constant. Identification happens based on the gradient of the different radii. If they are subsequently decreasing or not changing, these parts are treated as one curve. The entry curve will thus be annotated with the lowest speed allowed in the whole curve. If the radius increases, this is treated as the end of a connected curve. In that case, speed changes are allowed again. Based on the street shown in Figure 3, the connected curve would begin with point 0 and end with point 2. Afterwards, the radius increases and calculations do not influence the previous curve anymore. Speed at point 0 and point 2 is equal. In principle, it is shown that the proposed approach allows safe driving through curves remotely. This approach is additionally strengthened by regulations, that define allowed curve radii, etc., e.g. the German "Richtlinien für die Anlage von Landstraßen" [40].

4) Treatment of Packet Loss, Bandwidth and Jitter: The aforementioned approaches consider latency to be stable. Unfortunately, network parameters change frequently and jitter occurs. As shown by Davis et al. [17] and Liu et al. [41], constant delay can be handled better than variable one. Luckily jitter could be easily tackled by introducing buffers, e.g. as shown by d'Orey et al. in [42]. The buffer is set to pre-defined, measurement- and area-based values that lay above the expected real-world latency. Thus, jitter below the buffer will not influence the remote operator. Buffer might change during driving, but this will happen rarely compared to changes of raw jitter values. Thus, jitter can be overcome and is no showstopper. Besides latency, bandwidth and packet loss are important and have to be considered. To fit them into the proposed approach, mapping them to latency is crucial. The transmission of video-streams and other sensor data through the network is essential for the driver to sense the surrounding of the remote vehicle. If bandwidth decreases, raw video-streams and other data can not be transmitted anymore. To overcome this issue up to a certain level, the application of a compression algorithm with a higher compression rate is required, e.g. as shown by Kang et al. [3] for video compression. Higher compression typically requires more computational time [43]. Knowing the bandwidth of specific areas and the time compression will approximately take, bandwidth can be mapped to latency.

Usually packet loss is defined by a probabilistic factor, e.g. 0.01% packet loss during a measurement. To deal with packet loss, it is assumed that control- and environmental packets are sent frequently. Formula 7 can be used to approximate the additional required latency l_l for probabilistic packet loss. $1/T$ indicates the frequency packets are sent with, p_{loss} is the probability of packet loss and N is the roughly estimated number of packets sent. The final result will always be rounded up to a multiple of $1/T$.

$$l_l = \lceil \frac{\sum_{i=0}^{N-1} \frac{1}{T} * (p_{loss})^i}{1/T} \rceil * \frac{1}{T} \quad (7)$$

5) Putting it together: The first step is to define the origin and the destination of the vehicle. This could be directly at the beginning of an autonomous ride, or at the point in time when a remote driver has to take over. Defining origin and destination at the beginning of the ride makes sense, as route planning then can avoid areas with insufficient network performance. Otherwise it could happen, that network performance is too bad and remote takeover is impossible. Based on the predicted network performance in specific areas and the accordingly allowed and blocked streets, the route planning approach figures out a suitable route. For a first approach, this will happen by using already known algorithms (e.g. as described by Delling et al. [44]) to calculate the k fastest or shortest routes depending on the requirements and available resources. Based on the k routes and known network performance, allowed speeds as well as buffer sizes will be calculated. Buffer size t_b is calculated with Formula 8, adding up jitter j_b, compression c_b and packet loss p_b separately for each area.

$$t_b = j_b + c_b + p_b \quad (8)$$

The pre-calculated k routes will be augmented with the buffer-sizes. As safety is an important factor, the route with the smallest buffer size and accordingly the best network performance out of the k routes will be determined and selected as best route. This approach avoids dealing with complex heuristics, e.g. weighting distance/time and safety, which will be part of future work.

If Teleoperated Driving is required, a remote operator will take over control of the vehicle seeing driving relevant information, e.g. speed limitations, network parameters and curvature of upcoming curves. The displayed information will be influenced by the current probed network parameters and thus is updated permanently. Current allowed speed v_e can be below the previous calculated values, as it depends on curvature, stopping distance, time to vehicle ahead and current network parameters. It will be calculated by Formula 9.

$$v_e = min(v_{curve}, v_{stop}) \quad (9)$$

The vehicle is allowed to drive maximal either the speed allowed by curve or stopping distance. Additionally, this speed is only allowed if the time distance to a vehicle ahead t_s is above the minimal time distance t_{min}. Otherwise the vehicle will have to increase the safety distance. It can happen that the a route has to be changed dynamically during driving if network parameters are getting too bad on the selected route. A vehicle will query frequently if there are any changes. Calculations then will start again and suggest an alternative route. Jumps in latency, implied by changes of the buffer, will be announced early and speed suggestions will already consider them so that no unexpected changes will happen.

C. Limitations

The approach shown in this paper aims to present the conceptional idea and thus has its limitations. One major limitation is probing that only distinguishes between usual and unusual days. For a real-world system probing must be more accurate and frequent, e.g. by providing a better classification of network parameters based on the number of measurements, the timestamp of measurements, etc. Values might be above the calculated buffer and thus get dangerous for Teleoperated Driving. This can be overcome by various approaches, e.g. multi-path communication [2] or triggered safety stops, if network performance gets too poor. The approach of treating packet loss must be adjusted to treat key-frames if using a key-frame based compression, e.g. with Forward-Error-Correction (FEC). This can be done for example by extending Formula 7 with compression specific parameters. Route planning, so far, does not consider traffic jams or dynamic speed limitations during calculations, thus the selected route will be optimized in future. It can also be extended in future work by calculating a heuristic-based route considering weighted distance/time and network performance. This also applies to the avoidance of braking in curves. Driving might be slower than it has to be, as speed changes will not happen. Considering that braking slightly is allowed in curves, this can be optimized in future. Speed changes are treated as instant and constant, which is not the case in real-world driving. Adding deceleration and acceleration values will help here. The mathematical description of vehicle dynamics is kept simple in this paper. Formulas can easily be replaced by more complex ones considering environment and vehicle. In general, the approach only works within defined boundaries as video-stream quality at least requires a sufficient resolution and specific encoders may add too much latency [45].

IV. First Results in Real-World

A real-world calculation is presented to show a basic initial proof of the presented approach. The proposed route originates at Esplanade (top right, green marker) and ends at Probierlweg (bottom left, red marker) (Figure 4). Based on the introduced calculations, a non-latency influenced driver with assumed instant acceleration and deceleration (see limitations) could achieve an average speed of 44.73 km/h, which is indicated by the blue line in Figure 5. This average speed is claimed comparable to real-world drivings, as it could be approximated by driving the route in real-world with little traffic. Network measurements in the afternoon on usual days in the route's area revealed a constant coverage with LTE/LTE-Advanced, median latency of 55 ms in Round-Trip-Time (RTT), sufficient bandwidth so that only basic compression had to be applied and packet loss in the range of 0.003%. Thus, this area has sufficient network performance and is a potential route for a teleoperated vehicle. Considering these network parameters a total latency of 0.2

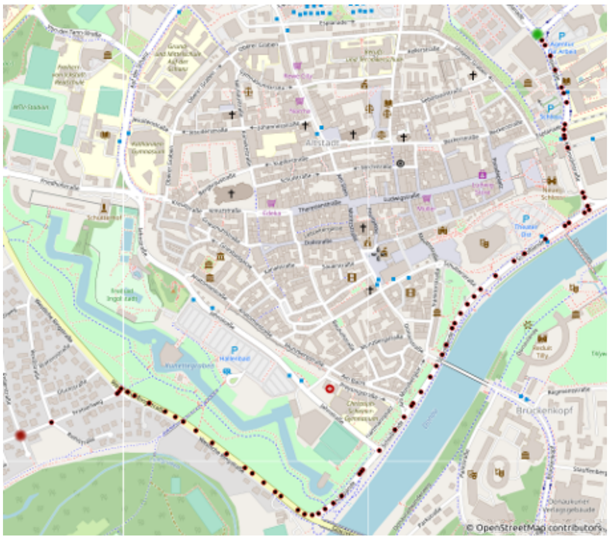


Fig. 4. Designated route from Esplanade (green marker) to Probierlweg (red marker), annotated with OSM-points (red markers with black surrounding) (from [27]).

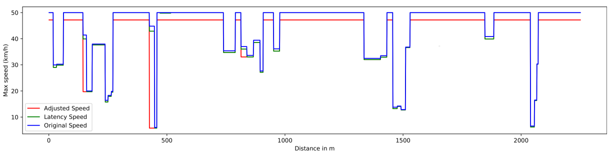


Fig. 5. Typical driving speeds on the selected route.

s, consisting of 55 ms in round-trip time, 20 ms based on packet loss calculations and about 125 ms of system latency, can be assumed. The new latency-based raw curve speeds are calculated and can be seen as green line in Figure 5. This new calculated speed is 44.56 km/h in average and only about 0.2 km/h below the unadjusted speed. Applying curve-smoothing and stopping, allowed speed slightly changes (red line in Figure 5) and the average driving speed decreases to approximately 42 km/h. This is a difference of about 3 km/h compared to the non-latency driving and thus perfectly suitable.

V. Conclusion and Future Work

With this paper a twofold approach for enabling Teleoperated Driving is proposed. It generally consists of route planning and driver augmentation. Based on previous and current network parameters, routes eligible for Teleoperated Driving are planned. Additionally, it has been shown that latency can be counteracted with speed adjustments. Packet loss and bandwidth within boundaries can be turned into latency values and thus used within the approach. Augmentation with relevant parameters will help the operator to drive safely. The twofold approach is suitable for Teleoperated Driving and might be used in future implementations as it shows, that Teleoperated Driving must not be stopped by latency, bandwidth or packet loss (at least within the previously defined boundaries). In future work, the presented approach will be improved. The route planning has to become smarter. It should not only be based on the best network performance of preselected routes, but also on the amount of other vehicles as well as on a balance between safety and comfort. The allowed speed calculations will be improved considering environmental conditions and speed changes over time. Further, it has to be investigated how classifying probes can be done efficiently, e.g. how to classify and fit best to the current driving situation. It is also important to test this approach extensively within real-world scenarios to identify drawbacks and see if it will have influence on traffic, e.g. slowing down other traffic participants.

In principle, the proposed approach works but needs improvements for applying it within real-world scenarios.

References

[1] On-Road Automated Driving (ORAD) committee, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE International Std., June 2018.

[2] M. Harris, "CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads," https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads, Jan. 2018, accessed on 28.11.2018.

[3] L. Kang, W. Zhao, B. Qi, and S. Banerjee, "Augmenting self-driving with remote control: Challenges and directions," in *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, ser. HotMobile '18. New York, NY, USA: ACM, 2018, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/3177102.3177104

[4] A. Davies, "The War to Remotely Control Self-Driving Cars Heats Up," https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars/, Mar. 2019, accessed on 04.04.2019.

[5] M. Sauter, *From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband, Third Edition*, 3rd ed. WILEY, 2017.

[6] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-advanced Pro and the Road to 5G*. Academic Press, 2016.

[7] A. F. Winfield, "Future Directions in Tele-operated Robotics," in *Telerobotic Applications*, 2000.

[8] T. Tang, F. Chucholowski, and M. Lienkamp, "Teleoperated driving basics and system design," *ATZ worldwide*, vol. 116, no. 2, pp. 16–19, Feb 2014. [Online]. Available: https://doi.org/10.1007/s38311-014-0018-1

[9] T. Tang, D. Soto-Setzke, C. Kohl, T. Köhn, J. Lohrer, and J. Betz, "EE-Architektur für mobile Dienste," *ATZextra*, vol. 19, no. 14, pp. 40–45, 2014. [Online]. Available: http://dx.doi.org/10.1365/s35778-014-1356-8

[10] S. Lichiardopol, "A survey on teleoperation," *Technische Universitat Eindhoven, DCT report*, 2007.

[11] S. Neumeier, N. Gay, C. Dannheim, and C. Facchi, "On the Way to Autonomous Vehicles - Teleoperated Driving," in *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*. Dortmund, Germany: VDE, 2018, pp. 1–6.

[12] S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp, "A System Design for Teleoperated Road Vehicles." in *ICINCO 2013 - Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, Reykjavik; Iceland, July 2013, pp. 231–238.

[13] X. Shen, Z. J. Chong, S. Pendleton, G. M. James Fu, B. Qin, E. Frazzoli, and M. H. Ang, *Teleoperation of On-Road Vehicles via Immersive Telepresence Using Off-the-shelf Components*. Cham: Springer International Publishing, 2016, pp. 1419–1433. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-08338-4_102

[14] F. Chucholowski, T. Tang, and M. Lienkamp, "Teleoperated Driving Robust and Secure Data Connections," *ATZelektronik worldwide*, vol. 9, no. 1, pp. 42–45, Feb 2014. [Online]. Available: https://doi.org/10.1365/s38314-014-0226-x

[15] S. I. M. T. Deutschland, "Deliverable D5.5 – Teil A TP5-Abschlussbericht – Teil A," http://www.simtd.de/index.dhtml/object.media/deDE/8154/CS/-/backup_publications/Projektergebnisse/simTD-TP5-

Abschlussbericht_Teil_A_Manteldokument_V10.pdf, June 2013, accessed on 05.12.2016.

[16] R. El Hattachi and J. Erfanian, "5G White Paper," NGMN Alliance, Tech. Rep., Feb. 2015.

[17] J. Davis, C. Smyth, and K. McDowell, "The Effects of Time Lag on Driving Performance and a Possible Mitigation," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 590–593, June 2010.

[18] F. E. Chucholowski, "Evaluation of Display Methods for Teleoperation of Road Vehicles," *Journal of Unmanned System Technology*, vol. 3, no. 3, pp. 80–85, 2016.

[19] A. Hosseini and M. Lienkamp, "Predictive safety based on track-before-detect for teleoperated driving through communication time delay," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. Gothenburg, Sweden: IEEE, June 2016, pp. 165–172.

[20] T. Tang, P. Vetter, M. Lienkamp, S. Finkl, and K. Figel, "Teleoperated Road Vehicles – The "Free Corridor" as a Safety Strategy Approach," in *2nd International Conference on Mechanical Design and Power Engineering, ICMDPE 2013*, Beijing; China, 2014, pp. 1399–1409.

[21] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 927–941, Oct 2007.

[22] R. Chellali and K. Baizid, "What maps and what displays for remote situation awareness and rov localization?" in *Human Interface and the Management of Information. Interacting with Information*, G. Salvendy and M. J. Smith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 364–372.

[23] A. Hosseini and M. Lienkamp, "Enhancing telepresence during the teleoperation of road vehicles using HMD-based mixed reality," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. Gothenburg, Sweden: IEEE, June 2016, pp. 1366–1373.

[24] J. Georg, J. Feiler, F. Diermeyer, and M. Lienkamp, "Teleoperated driving, a key technology for automated driving? comparison of actual test drives with a head mounted display and conventional monitors*," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI, USA: IEEE, Nov 2018, pp. 3403–3408.

[25] S. Neumeier, E. Walelgne, V. Bajpai, J. Ott, and C. Facchi, "Measuring the Feasibility of Teleoperated Driving in Mobile Networks," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*. Paris, France: IEEE, June 2019, pp. 113–120.

[26] OpenCelliD, "OpenCelliD - Largest Open Database of Cell Towers & Geolocation - by Unwired Labs," https://opencellid.org/, accessed on 10.03.2019.

[27] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[28] D. Beisel, R. Hosse, and E. Schnieder, "Modellbasierte Analyse der Auslegung von Assistenzsystemen," *ATZ - Automobiltechnische Zeitschrift*, vol. 115, no. 1, pp. 76–82, Jan 2013. [Online]. Available: https://doi.org/10.1007/s35148-013-0018-2

[29] K. D. Kusano, R. Chen, J. Montgomery, and H. C. Gabler, "Population distributions of time to collision at brake application during car following from naturalistic driving data," *Journal of Safety Research*, vol. 54, pp. 95.e29 – 104, 2015, strategic Highway Research Program (SHRP 2) and Special Issue: Fourth International Symposium on Naturalistic Driving Research. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022437515000432

[30] B. Sidaway, M. Fairweather, H. Sekiya, and J. Mcnitt-Gray, "Time-to-collision estimation in a simulated driving task," *Human Factors*, vol. 38, no. 1, pp. 101–113, 1996. [Online]. Available: https://doi.org/10.1518/001872096778940813

[31] W. Spielmann and M. Reuter, "Bremswege als vergleichsgröße zwischen bremsanlagen," *ATZ - Automobiltechnische Zeitschrift*, vol. 104, no. 5, pp. 464–472, May 2002. [Online]. Available: https://doi.org/10.1007/BF03224409

[32] F. Kost, *Grundlagen der Fahrphysik*. Wiesbaden: Vieweg+Teubner Verlag, 2010, pp. 20–35. [Online]. Available: https://doi.org/10.1007/978-3-8348-9714-5_2

[33] K. Reif and K.-H. Dietsche, *Kraftfahrtechnisches Taschenbuch*. Springer-Verlag, 2019.

[34] D. Schramm, M. Hiller, and R. Bardini, *Single Track Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 223–253. [Online]. Available: https://doi.org/10.1007/978-3-540-36045-2_10

[35] M. Harrer and P. Pfeffer, *Fahrdynamische Grundlagen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2013, pp. 77–100. [Online]. Available: https://doi.org/10.1007/978-3-658-00977-9_5

[36] C. F. F. Karney, "Algorithms for geodesics," *Journal of Geodesy*, vol. 87, no. 1, pp. 43–55, Jan 2013. [Online]. Available: https://doi.org/10.1007/s00190-012-0578-z

[37] K. Kendig, "Is a 2000-year-old formula still keeping some secrets?" *The American Mathematical Monthly*, vol. 107, no. 5, pp. 402–415, 2000.

[38] M. Mitschke and H. Wallentowitz, *Kreisfahrt bei konstanter Fahrgeschwindigkeit*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014, pp. 625–655. [Online]. Available: https://doi.org/10.1007/978-3-658-05068-9_21

[39] L. L. Schumaker, *Spline Functions: Computational Methods*. SIAM, 2015, vol. 142.

[40] Forschungsgesellschaft für Straßen- und Verkehrswesen, Arbeitsgruppe Straßenentwurf, "Richtlinien für die Anlage von Landstraßen : RAL," Köln, 2012.

[41] R. Liu, D. Kwak, S. Devarakonda, K. Bekris, and L. Iftode, "Investigating remote driving over the lte network," in *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '17. New York, NY, USA: ACM, 2017, pp. 264–269. [Online]. Available: http://doi.acm.org/10.1145/3122986.3123008

[42] P. M. d'Orey, A. Hosseini, J. Azevedo, F. Diermeyer, M. Ferreira, and M. Lienkamp, "Hail-a-Drone: Enabling teleoperated taxi fleets," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. Gothenburg, Sweden: IEEE, June 2016, pp. 774–781.

[43] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders," in *2013 Picture Coding Symposium (PCS)*. San Jose, CA, USA: IEEE, Dec 2013, pp. 394–397.

[44] D. Delling, P. Sanders, D. Schultes, and D. Wagner, *Engineering Route Planning Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 117–139. [Online]. Available: https://doi.org/10.1007/978-3-642-02094-0_7

[45] N. Barman and M. G. Martini, "H.264/MPEG-AVC, H.265/MPEG-HEVC and VP9 codec comparison for live gaming video streaming," in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. Erfurt, Germany: IEEE, May 2017, pp. 1–6.

Publication [V]: Yet Another Driving Simulator OpenROUTS3D: The Driving Simulator for Teleoperated Driving

[**V**] Stefan Neumeier, Michael Höpp, and Christian Facchi. “Yet Another Driving Simulator OpenROUTS3D: The Driving Simulator for Teleoperated Driving”. In: *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. Graz, Austria: IEEE, Nov. 2019

This paper introduces the driving simulator OpenROUTS3D. We first discuss why the development of a new driving simulator was required as the existing ones did not provide the required functionality or were too expensive. Afterwards we introduce its features, which are especially important for teleoperated driving. The driving simulator itself is based on the Unity platform and as such mainly developed in C#. We always developed it with the sake of providing the final driving simulator as Open source to the community[1]. One of the main features is the dynamic creation of maps based on Simulation of Urban MObility (SUMO) data, which is also responsible for creating artificial traffic, or the direct import of Unity assets. In order to allow for the investigation of teleoperated driving scenarios, it is important to include network parameters such as latency and bandwidth in the simulation. Both can be defined using the driving simulator, e. g., introducing different types of latency and changing the displayed image quality. In order to allow for a detailed analysis of drivings, a logging system is responsible for storing vehicle and environmental data frequently. Additionally, this data can be transmitted through the network to monitor the driving performance in real-time on another systems. A recorded ride can be replayed to analyze specific situations by exactly seeing what the driver has seen or by changing the view arbitrarily. Another feature is the implementation of different sensors that provide features such as LiDAR or video camera. Within the multiplayer mode it is possible to control several remote-controlled vehicles by multiple teleoperators to allow the simulation of driving scenarios with greater complexity. To conduct user studies that minimize the distraction of participants, we implemented a user study mode. This mode allows to conduct user studies with minimal interaction between the

[1]https://github.com/sneumeier/OpenROUTS3D

participant and the investigator. It allows to define a whole user study including questionnaire, driving, introduction, etc. This study then will run seamlessly being fully controlled by the participant. Finally OpenROUTS3D is meant to be flexible and easily extensible. As such, it offers an addon system that provides an easy replacement of components of the driving simulator. It also allows to add further functionality without the need of changing the source code of the driving simulator itself.

Due to the huge effort on developing such a complex driving simulator, it was not developed on my own. Instead it was developed and improved by two student-projects and supported by multiple working students, which I guided. However, my contribution, besides the major part in writing of the paper, consisted of defining the software architecture and leading the way of the development. This included the first idea and the initial definition of the software system. As such, I analyzed the features of existing driving simulators and derived requirements to fit the driving simulator for the needs of teleoperated driving. Additionally, I kept track of the development and was responsible for reviewing the code and guiding the direction of development, e. g., defining and selecting features. In addition I supported in development when necessary. The writing of the paper was initiated by me and happened in close exchange with the co-authors. I also mainly took care of addressing reviewers' comments.

Yet Another Driving Simulator OpenROUTS3D: The Driving Simulator for Teleoperated Driving

Conference Proceedings:
2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)
Author: [::Stefan::] [::Neumeier::]; Michael Höpp; Christian Facchi
Publisher: IEEE
Date: 4-8 Nov. 2019

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Yet Another Driving Simulator OpenROUTS3D: The Driving Simulator for Teleoperated Driving

Stefan Neumeier
Technische Hochschule Ingolstadt
Research Centre
Email: stefan.neumeier@thi.de

Michael Höpp
Technische Hochschule Ingolstadt
Email: michael@hoepp-in.de

Christian Facchi
Technische Hochschule Ingolstadt
Research Centre
Email: christian.facchi@thi.de

***Abstract*—Numerous issues on the way to autonomous vehicles have already been solved. Nevertheless, there are further problems preventing the introduction of autonomous driving features of higher SAE levels. Remote control of vehicles by human operators located in dedicated operation centers, Teleoperated Driving, can help to overcome the problems of autonomous driving. To enable functional Teleoperated Driving, existent network technology has to be utilized. These cellular networks suffer from variable performance. However, testing Teleoperated Driving and its algorithms in real-world scenarios is costly and potentially dangerous. Virtual testing is an approach to mitigate these obstacles. This paper introduces OpenROUTS3D, an open-source driving simulator initially developed for Teleoperated Driving, but expandable to various use cases.**

I. Introduction

A multitude of problems associated with driver assistant systems have already been solved, but further improvements are required to achieve levels beyond SAE [4] level 3. If sensor or hardware errors occur [16] or a situation is too complex for autonomous systems [13], a human remote operator located in a dedicated operation center could act as fallback solution. This is called Teleoperated Driving and currently developed and partly forced by law, e.g. in California for testing driverless vehicles [8]. To use Teleoperated Driving sensibly, it is of advantage to use already existing infrastructure such as cellular networks. With modern technologies like LTE/LTE-Advanced [25] and 5G [7], cellular networks are able to provide sufficient network performance. However, these networks suffer from various issues regarding bandwidth, latency and packet loss, preventing a comprehensive use of Teleoperated Driving. Overcoming them requires the development of specialized algorithms and skilled remote operators combined with extensive testing. This can not be achieved solely by real-world development and testing. Therefore, virtual test drives are unavoidable. Focusing on different aspects or being too costly, existing driving simulators are not suitable for Teleoperated Driving. Therefore, a flexible, extensible, free and easy-to-use driving simulator is required. Addressing these objectives, this paper introduces OpenROUTS3D (**Open R**ealtime **O**SM- and **U**nity-based **T**raffic **S**imulator **3D**), a driving simulator developed for Teleoperated Driving. It has been extended and is now an universal tool applicable for developing and testing various vehicular applications. This paper discusses why a specialized driving simulator for Teleoperated Driving is required, which functionality is available in OpenROUTS3D and which software-design decisions were made.

II. Related Work

A lot of previous research and development in the area of Teleoperated Driving has been carried out already. Windfield [35] describes the three basic components a teleoperated system consists of as: remote vehicle, operator's workplace and their connection. Basic research on teleoperated vehicles has been carried out for example by Tang et al. in [28] and [29]. Lichiardopol did a survey on Teleoperated Driving in [17]. Exemplary setups of Teleoperated Driving systems are shown by Neumeier et al. [20], Gnatzig et al. [12] and Shen et al. [27]. Teleoperated Driving was already simulated in previous research, e.g. in [14] (SILAB) or [15] and [3] (DYNA4), but their setups are not publicly accessible.

A proprietary software for virtual test drives is CarMaker [1]. It can be used in all steps of the development process and offers the opportunity to transfer real-world scenarios to a virtual world and test them with high level of details. Automotive Data and Time-Triggered Framework (ADTF) [11] is a framework that can support the development process of autonomous software. It is a tool for "the development, validation, visualization and test of driver assistance and automated driving features that includes the latest technology" [11]. AVL Cruise [2] is a vehicle driveline simulation, which can be used for powertrain analysis and within Hardware-in-the-Loop (HiL) and Software-in-the-Loop (SiL) testbeds. These tools are powerful but proprietary and potentially expensive.

With Open Source Driving Simulation (OpenDS) [19] an open-source, GNU GPL-licensed and java-based driving simulator exists. It is platform independent, consists of analysis and simulator components and provides features like traffic simulation, highly detailed cars and different weather conditions. Thus, it can be used for different types of applications. CARLA is a MIT-licensed open-source simulator for research in the area of autonomous driving [9]. With a flexible API, ROS integration, map generation based on OpenDrive [10] and sensor simulation, it is designed to support all steps in the development of autonomous driving systems. It also provides digital assets to support the development process. Another driving simulator available as open-source software under the MIT license is AirSim developed by Microsoft [26]. It is based on the Unreal engine, but an experimental Unity release is available. The goal of AirSim is to provide a platform for conducting experiments with AI, e.g. computer vision. Hence, an API for retrieving data and controlling vehicles exists. Udacity's Self-Driving Car Simulator [34] was developed to train cars to navigate through courses based on machine's decisions. It is based on Unity and available as open-source software under the MIT license. Multi-Agent DRiving Simulator (MADRaS) [24] is a multi-agent version of the GPL-licensed TORCS racing simulator [36], which can be used as research platform. MADRaS extends TORCS and offers the opportunity to control multiple vehicles and thus allows to simulate complex scenarios. Deepdrive [23] is an end-to-end simulator for autonomous vehicles. It is based on the Unreal engine and licensed under the MIT license. It offers a fast way of getting started with experiments in the area of self-driving vehicles. Generally speaking, available open-source software is not focused on Teleoperated Driving and would mean a lot of work to integrate required features, without knowing if it pays off. Thus, it was decided to develop a driving simulator from scratch, mainly focused on Teleoperated Driving.

III. Features

Starting with the development of OpenROUTS3D, focus was on a driving simulator fitting the specific needs of Teleoperated Driving. Thereon, OpenROUTS3D has been improved to support further and more general use cases. In addition to the driving simulator itself, OpenROUTS3D currently ships with a Python-based supporting tool facilitating detailed analysis of drives by considering all log-files.

A. General Information

OpenROUTS3D is based on the Unity platform [30] and thus mainly developed in C#. It is publicly available at github[1] as open-source software licensed under GNU GPLv3 providing the option of fully customizing for specific needs. All included models are either self-developed or compatible with the GPLv3 license and delineated appropriately. The driving simulator's basic system architecture, shown in Figure 1, consists of OpenROUTS3D with internal modules and Simulation of Urban MObility (SUMO) [18] with the Traffic Control Interface (TraCI) used for traffic simulation. Input files are used for map creation, user study execution or adding features. Output mainly consists of log-files. Steering and displaying are treated separately. OpenROUTS3D can be used with all major operating systems including Windows, Linux and macOS. Technology of choice is Unity allowing to create a modular, open, object-oriented driving simulation that runs

[1]https://github.com/sneumeier/OpenROUTS3D

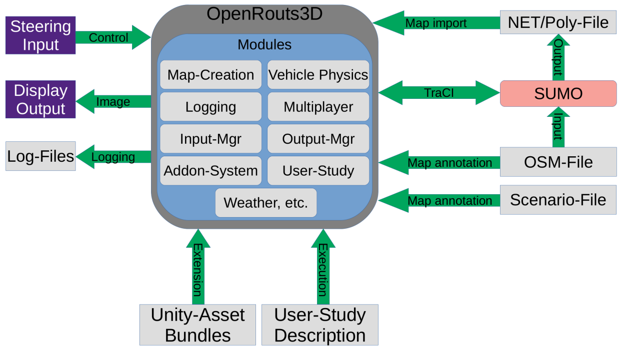


Fig. 1: Basic system architecture of OpenROUTS3D.

on various platforms. The driving simulator was designed to enable a quick start and thus already offers multiple exemplary scenarios and different vehicle models for artificial traffic. However, so far there is only one vehicle configured with a cockpit to act as ego-vehicle. Its driving physics are based on the Randomation-Vehicle-Physics project [6], offering a huge amount of options for configuring vehicle's behavior, e.g. suspension, engine, etc. Although it is easy to get started with, its modular architecture allows the driving simulator to provide a high degree of flexibility and extensibility. Therefore, it can be used in various scenarios. OpenROUTS3D does not require particular powerful hardware and thus can be run even on non-gaming notebook systems. It was successfully tested on a typical business notebook equipped with an Intel Core i5 CPU and an integrated Intel HD graphics 620 card.

B. Map and Artificial Traffic Creation

Maps for OpenROUTS3D can be created in multiple ways utilizing SUMO, Unity or a combination of both. The straightforward way is using a SUMO-road network. The driving simulator interprets the road topology and builds streets, intersections, etc. Streets are created using spline interpolation, while intersections are based on SUMO's shape information. Traffic lights and speed limitations are created if required information is available. If there are any corresponding information regarding polygons in a polygon-file, e.g. shape information of buildings or forests, they will be used to create objects accordingly. Further information provided by OpenStreetMap (OSM) files [22], will be considered if available. OSM specific parameters are thereby available to the driving simulator. This allows to take different types of surfaces and street-names amongst others into account. If such type of information is missing because either if there is no OSM-file or there is no information about it, surfaces and building heights are randomly generated. SUMO-based maps can be created either automatically or by hand (left side of Figure 2). Using OSM-maps as base helps to obtain information about polygons, building heights, street limitations, etc. This automated approach is useful if one wants to create a city scenario for instance. The SUMO tool *NETCONVERT* together with an OSM-map or the *osmWebWizard* script, which allows the comfortable creation of a complete scenario, enable an

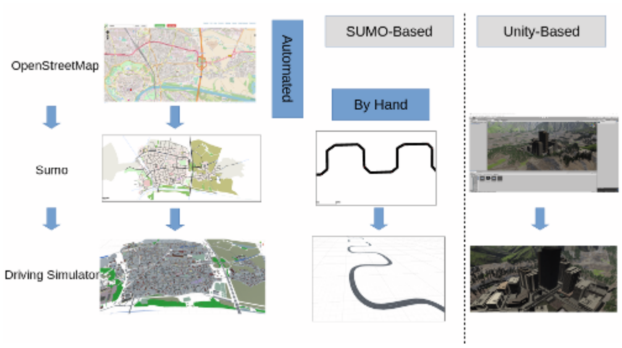
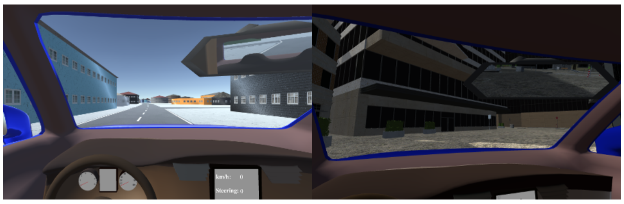


Fig. 2: Automated and manual map creation for OpenROUTS3D based on SUMO next to Unity-based map creation.



(a) Auto-Generated (b) AirSim

Fig. 3: Screenshots of auto-generated (a) and Unity-Plugin (b) based maps.

automated creating. The manual way of map creation is based on designing the street-network by hand. This can be useful in situations where no OSM-files exist or are unsuitable. Manual maps can either be created with *NETEDIT*, or with a SUMO-independent tool like JOSM [5]. The second way of creating SUMO-independent maps is the use of OpenROUTS3D's Addon-system to import Unity-based assets, depicted with the Windridge City-Asset [33], an environment for AirSim [26] on Unity, shown on the right side of Figure 2. Every type of Unity-based asset, e.g. the one of Microsoft AirSim [26], can be loaded and used within the driving simulator. This allows the creation of highly detailed environments, which can be important for computer vision topics. To allow SUMO-based traffic on such scenarios, a combination of SUMO and Unity-assets is required. In such an approach, SUMO is used to create the basic street network, while the Unity-based objects have to be placed accordingly. This does not necessarily mean that SUMO draws the streets. So far this combination is only possible with flat areas at the same height above sea level as SUMO maps are always treated as flat. An example of SUMO-based maps and imported Unity-based maps can be seen in Figure 3. On the left side an automatically created plain SUMO-map with OSM-based information is shown, whereas the highly detailed map of Windridge City [33] is shown on the right. Auto-generated maps are suitable for most use cases, but if high detailed-models are required they could be added using OpenROUTS3D's Addon-system.

The easiest way to create real-world-like behaving traffic is the use of SUMO. Via TraCI it is possible to obtain information about traffic lights and non-ego vehicles and transfer their continuously updated states/positions to the driving simulator. This coupling can be seen in Figure 4, where vehicles and traffic lights are synchronized between SUMO and the driving simulator. Due to the fact that TraCI is bidirectional, SUMO can react on user's behavior adjusting the artificial traffic. Artificial traffic right now consists of vehicles only, but could be extended to cyclists, pedestrians, railways, etc. The TraCI interface also allows to control the SUMO simulation and thus enables influencing the simulation, e.g. removing or adding vehicles. Another way, best suited for scenarios in which something unpredictable should happen, is the use of the Addon-system. This allows the creation of arbitrary objects that follow a specific predefined movement pattern. For example, it is possible to create suddenly appearing crossing traffic on an intersection. This can be used to test evasive maneuvers in Teleoperated Driving.

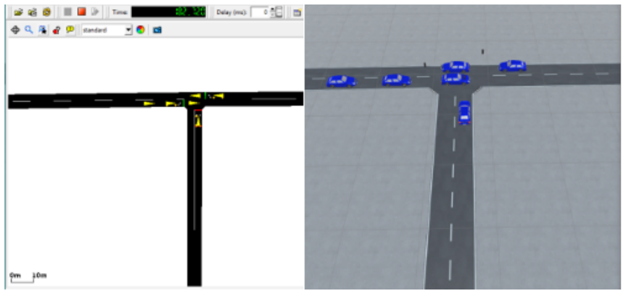


Fig. 4: Coupling of SUMO and OpenROUTS3D.

Fig. 5: Rain and snow in OpenROUTS3D.

In addition to the creation of streets and environment, OpenROUTS3D allows the configuration of different weather conditions. These conditions can be defined either globally or individually configured for specific areas. However, weather conditions do not influence driving physics yet. Weather configuration is possible by using one of the presets, or configuring specific conditions individually. A screenshot of a configuration with snow and rain can be seen in Figure 5.

Finally, OpenROUTS3D allows the creation of XML-based scenarios within the driving simulator. This enables scenario-specific adjustments like pylon placing, defining areas with different Quality of Service parameters or defining the spawn position of the ego-vehicle. The scenarios consist of a serialized XML-file that contains all information that should be loaded appropriately and can be created within the driving

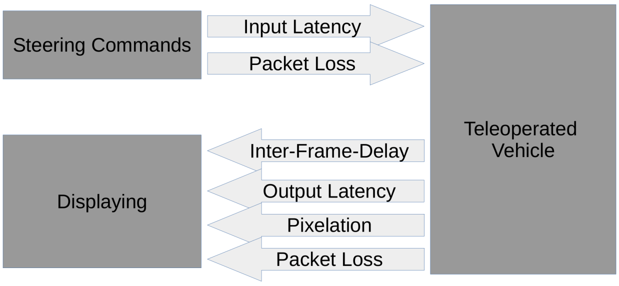
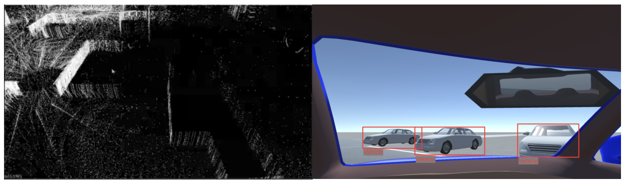


Fig. 6: OpenROUTS3D's adjustable Input/Output parameters.



(a) LiDAR (b) Camera

Fig. 7: LiDAR (a) and camera with object-classification (b) in OpenROUTS3D.

simulator or with an external tool.

C. Input and Output

The general concept behind input and output handling is designed to fit the needs of Teleoperated Driving, but can be used for any other vehicular application. The input-manager can deal with different types of controllers. Thus, OpenROUTS3D can be used with a keyboard, but also supports steering and pedals as well as other gamepads. The output can either be a (multi-)monitor setup or a head-mounted virtual-reality headset. Input and output buffers are used to enable artificial latency. To simulate different types of resolutions and frame rates, the output further supports pixelation and inter-frame delay. Packet loss is designed to randomly discard packets based on configurable probabilistic distribution. Input delay is realized with an adjustable buffer. It can be configured to only consider control inputs and handle scenario specific inputs like scene skipping, etc. differently. Output delay, inter-frame delay and pixelation are realized by pre-rendering an image and displaying it with respect to the configured delay. All these influencing factors for Teleoperated Driving, shown by Figure 6, can be configured statically, which means that they are consistent and not changing on a map, or dynamically. A dynamic configuration allows defining different area-dependent parameters to simulate real-world behavior.

D. Logging System and Replay Feature

Besides the TCP-based live-stream of driving parameters, OpenROUTS3D writes three different types of log-files, generally consisting of drive-logs, collision logs and results of questionnaires. The drive-log is CSV-based and stores different information regarding the current state of the vehicle and its environment every frame. Among others this includes the exact ego-vehicle's position, its heading, distance covered, the inputs of pedals/steering and the applied input/output parameters allowing for calculations such as Standard Deviation of Lateral Position, etc. Collision logs are XML-style consisting of information about collisions and their impact. Answers to questionnaires are stored in a separate third key-value-pair-like XML-file with support for multi-select and free-text answers. Together, these files allow for a detailed analysis. To remotely inspect the driving of a participant in real-time, the drive-log is additionally streamed using TCP.

Based on the drive-log the option to replay a completed drive exists. Replay does not cover vehicle physics calculations. Instead it takes the values stored in the log-file and updates the vehicle's position as well supporting parameters appropriately applying interpolation. Replay-speed can be changed dynamically. Therefore, it is possible to fast scroll to an interesting situation and inspect this situation in slow-motion. The basic idea behind such a feature is the visual analysis of drives to inspect specific parts of it, e.g. to visually analyze drives of an user study. Furthermore, this feature is useful for applications like LiDAR ray-calculations that drastically reduce frame rate. Instead of appending such a feature to a human-controlled drive, it is possible to conduct the drive normally and enable all required processing steps in the replay. This allows the application of resource consuming algorithms keeping real-time performance during driving.

E. Simulation of Sensors

OpenROUTS3D allows the implementation of specific sensors. Currently, there are two types of sensors implemented: a LiDAR sensor and a camera for object classification. Images of both can be seen in Figure 7. The LiDAR sensor creates point-cloud data based on ray-casting, which can be processed further with different tools. It is configurable regarding its columns, rows and angle. The camera detects configured objects in its line-of-sight and stores a screenshot along with their positions and the object type in an external file. Within the simulator the object type is known and thus can be used to create a data-set suitable for machine learning applications like traffic sign recognition. Cameras can be configured to focus on specific spots, have specific resolutions, etc. For Teleoperated Driving this allows the sensor-based testing of supporting algorithms, e.g. lane keeping assistant, etc.

F. Multiplayer

To allow driving with more than one user, OpenROUTS3D includes a multiplayer mode, where it is possible to drive together with multiple users via network connection. This can be used to create scenarios with more than one human participant, e.g. one could drive an emergency vehicle and others drive conventional vehicles. In multiplayer-mode, one system acts as host, while arbitrary clients can connect to it. Positions of the vehicles are distributed to all nodes, allowing a fluent driving. During the initial handshake, the

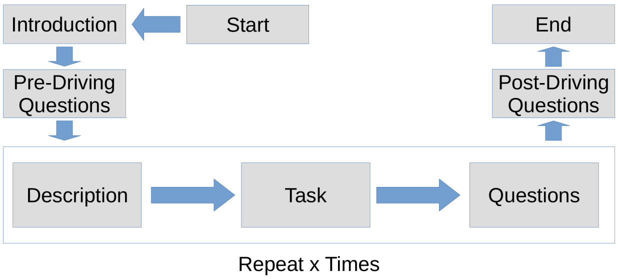


Fig. 8: Example study-circle.

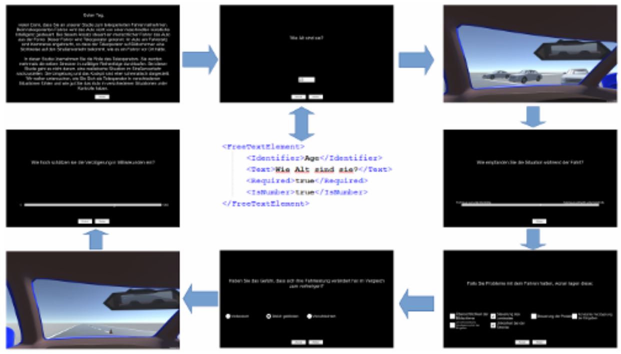


Fig. 9: Screenshots of an exemplary user study and a xml-based question within OpenROUTS3D.

host determines which scenario is loaded and provides basic information about it. The scene itself is not shared via network and thus needs to be present on all clients. Additionally, each client must have the same addons enabled as the host. Driving physics of ego-vehicles are calculated locally - merely resulting velocities, positions, rotations, etc. are shared via network. Thus, collisions between player-controlled vehicles at high latencies can behave differently on each individual system. The multiplayer-mode allows to drive multiple vehicles teleoperated and see how they interact, which is useful to test the influence of multiple teleoperated vehicles.

G. User Study Mode

Besides the development of technical aspects, research in autonomous and teleoperated driving focuses on human factors, usually determined by user studies. To address this, OpenROUTS3D provides an user study mode, that allows the creation and execution of different types of user studies. User studies typically consist of task description, driving scenarios and corresponding questionnaires. To avoid interruption or distraction of participants by an investigator, a seamless transition between driving tasks, descriptions and questionnaire is crucial. Therefore, the driving simulator allows the creation of fully-integrated studies, where participants do not require interactions with investigators. As the user study mode allows adding further instructions, a potential study-cycle could look like the one shown in Figure 8. The simulation starts, provides basic description and shows a basic questionnaire. Afterwards the first driving task is explained, the task starts and subsequently there is a questionnaire. This may be repeated several times until the study ends with a final questionnaire. The sequence and questionnaires of an user study can be described by XML-files as shown in Figure 9. Predefined types like single-select, multi-select, slider-input, free-text, etc. facilitate a fast creation of questionnaires. This allows a dynamic and tool-based creation of different scenarios for multiple participants. Driving tasks can be configured to stop automatically after a specific time-period, reaching a pre-defined area or by an action of the participant, e.g. pushing a button.

H. Addon-System

The Addon-system allows to add new objects and replace existing objects and Unity Prefabs [32], without the need of recompiling OpenROUTS3D. Addons can be installed by putting them in a specific folder, allowing easy sharing and executing. They are listed in the addon-section of the driving simulator's main menu and can be enabled or disabled by clicking. Addons for OpenROUTS3D are Unity asset-bundles with a specific structure. One object in such an asset-bundle has to contain a MonoBehaviour [31] script that provides certain attributes like an icon and load priority of the addon. This object serves as root object and as parent for all game objects that modify the simulator when the addon is enabled. Other non-game-object content like materials, icons or shaders, is deposited inside the asset-bundle and can be referenced by the previously mentioned game objects. All asset bundles are loaded and opened at application start to fill the addon-section of the main menu, yet only the ones toggled on are handled when starting a driving scene. Child objects below the addon root are handled differently depending on their scripts. They can be added to the simulation as child of an existing object (e.g. UI element that displays passed time), replace existing objects (e.g. alternative player vehicle model) or replace/add prefabs (e.g. alternative tree models or models for SUMO cars). Asset-bundles contain no information about the scene they are loaded into. References to objects in the simulation are passed as strings. Preset objects in the driving simulation, like the player car or the UI root, contain a script registering them as "anchor" with a specific string upon loading the simulation. The string references inside addons correspond to those anchor strings and are resolved before the first frame is rendered. OpenROUTS3D is shipped with a basic set of addons consisting of example addons and feature-adding addons like disabling camera movement, displaying a timer, randomized vehicles, randomized wall textures and replay of logs amongst others.

IV. Conclusion and Future Work

OpenROUTS3D is an Unity-based open-source driving simulator licensed under GNU GPLv3. Due to its versatility the use cases of the driving simulator are manifold. Its first use was during a driver study with about 30 participants [21]. Trying to analyze the impact of latency on driving performance, participants had to drive different constant and variable latencies on various scenarios like parking and driving zigzag between pylons. This user study revealed that the driving simulator is adjustable, robust, latency is applied correctly and its results can be analyzed easily. Furthermore, during the three weeks of the user study no failures occurred and participants liked the driving behavior. The driving simulator is further used in exhibitions to allow individuals to experience the challenges of Teleoperated Driving. OpenROUTS3D offers different ways of map and artificial traffic generation synchronizable with SUMO. Teleoperated Driving can be tested by configuring different Quality of Service aspects. Defining specific weather and different scenarios allows the creation of highly focused tracks, e.g. handling parkours. With the user study mode it is possible to define and conduct large user studies in an easy and straightforward way. Its features make it predominantly focusing on use cases regarding Teleoperated Driving, but its applications are not limited to it. Currently, development focuses on weather-influenced driving physic and improved treatment of height information. In future it is planned to improve the driving simulator to support a broader range of applications, e.g. integrating pedestrians and cyclists. In general it can be said, that OpenROUTS3D is an open, flexible and easy to use driving simulator that is well suited for Teleoperated Driving, but can be used for multiple applications.

References

[1] IPG Automotive. CarMaker — IPG Automotive. https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/. Accessed on 10.04.2019.

[2] AVL. AVL CRUISE. https://www.avl.com/cruise. Accessed on 10.04.2019.

[3] Frederic Emanuel Chucholowski. Evaluation of Display Methods for Teleoperation of Road Vehicles. *Journal of Unmanned System Technology*, 3(3):80–85, 2016.

[4] On-Road Automated Driving (ORAD) committee. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. Technical report, SAE International, January 2014.

[5] OpenStreetMap contributors. JOSM. https://josm.openstreetmap.de/. Accessed on 04.04.2019.

[6] Justin Couch. Randomation Vehicle Physics. https://github.com/JustInvoke/Randomation-Vehicle-Physics. Accessed on 20.03.2019.

[7] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G, LTE-advanced Pro and the Road to 5G*. Academic Press, third edition, 2016.

[8] Alex Davies. The War to Remotely Control Self-Driving Cars Heats Up. https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars/, March 2019. Accessed on 04.04.2019.

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[10] Marius Dupuis et al. Opendrive format specification. *VIRES Simulationstechnologie GmbH*, 2010.

[11] Elektrobit. EB Assist ADTF - Elektrobit. https://www.elektrobit.com/products/automated-driving/eb-assist/adtf/. Accessed on 10.04.2019.

[12] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. A System Design for Teleoperated Road Vehicles. In *ICINCO 2013 - Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, pages 231–238, Reykjavik; Iceland, July 2013.

[13] Mark Harris. CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads. https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads, January 2018. Accessed on 28.11.2018.

[14] A. Hosseini and M. Lienkamp. Enhancing telepresence during the teleoperation of road vehicles using HMD-based mixed reality. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1366–1373, Gothenburg, Sweden, June 2016. IEEE.

[15] A. Hosseini, F. Richthammer, and M. Lienkamp. Predictive Haptic Feedback for Safe Lateral Control of Teleoperated Road Vehicles in Urban Areas. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–7, Nanjing, China, May 2016. IEEE.

[16] Lei Kang, Wei Zhao, Bozhao Qi, and Suman Banerjee. Augmenting self-driving with remote control: Challenges and directions. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, HotMobile '18, pages 19–24, New York, NY, USA, 2018. ACM.

[17] S Lichiardopol. A survey on teleoperation. *Technische Universitat Eindhoven, DCT report*, 2007.

[18] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, Maui, HI, USA, 2018. IEEE.

[19] Rafael Math, Angela Mahr, Mohammad M Moniri, and Christian Müller. OpenDS: A new open-source driving simulator for research. In *AmE 2013 - Automotive meets Electronics*, 2013.

[20] Stefan Neumeier, Nicolas Gay, Clemens Dannheim, and Christian Facchi. On the Way to Autonomous Vehicles - Teleoperated Driving. In *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*, pages 1–6, Dortmund, Germany, 2018. VDE.

[21] Stefan Neumeier, Philipp Wintersberger, Anna-Katharina Frison, Armin Becher, Christian Facchi, and Andreas Riener. Teleoperation: The holy grail to solve problems of automated driving? sure, but latency matters. In *Proceedings of the 11th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '19, Utrecht, Netherlands, 2019. ACM. (Accepted Paper).

[22] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[23] Craig Quiter and Maik Ernst. deepdrive/deepdrive: 2.0, March 2018.

[24] Anirban Santara. MADRaS: A Multi-Agent DRiving Simulator. https://software.intel.com/en-us/articles/madras-a-multi-agent-driving-simulator, Apr 2018. Accessed on 10.04.2019.

[25] Martin Sauter. *From GSM to LTE-Advanced Pro and 5G: An Introduction to Mobile Networks and Mobile Broadband*. WILEY, third edition, 2017.

[26] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[27] Xiaotong Shen, Zhuang Jie Chong, Scott Pendleton, Guo Ming James Fu, Baoxing Qin, Emilio Frazzoli, and Marcelo H. Ang. *Teleoperation of On-Road Vehicles via Immersive Telepresence Using Off-the-shelf Components*, pages 1419–1433. Springer International Publishing, Cham, 2016.

[28] Tito Tang, Frederic Chucholowski, and Markus Lienkamp. Teleoperated driving basics and system design. *ATZ worldwide*, 116(2):16–19, Feb 2014.

[29] Tito Tang, David Soto-Setzke, Christopher Kohl, Thomas Köhn, Jürgen Lohrer, and Johannes Betz. EE-Architektur für mobile Dienste. *ATZextra*, 19(14):40–45, 2014.

[30] Unity Technologies. Unity. https://unity.com/. Accessed on 11.04.2019.

[31] Unity Technologies. MonoBehaviour. https://docs.unity3d.com/ScriptReference/MonoBehaviour.html, Apr 2019. Accessed on 11.04.2019.

[32] Unity Technologies. Prefabs. https://docs.unity3d.com/Manual/Prefabs.html, Apr 2019. Accessed on 11.04.2019.

[33] Unity Technologies. Windridge city. https://assetstore.unity.com/packages/3d/environments/roadways/windridge-city-132222, January 2019. Accessed on 05.03.2019.

[34] Udacity. Udacity's Self-Driving Car Simulator. https://github.com/udacity/self-driving-car-sim. Accessed on 10.04.2019.

[35] Alan Ft Winfield. Future Directions in Tele-operated Robotics. In *Telerobotic Applications*. 2000.

[36] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. Torcs, the open racing car simulator. *Software available at http://torcs. sourceforge. net*, 4(6), 2000.

Publication [VIII]: Teleoperation - The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters

[**VIII**] Stefan Neumeier, Philipp Wintersberger, Anna-Katharina Frison, Armin Becher, Christian Facchi, and Andreas Riener. "Teleoperation: The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters". In: *Proceedings of the 11th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '19. Utrecht, Netherlands: ACM, Sept. 2019. DOI: 10.1145/3342197.3344534. URL: https://doi.org/10.1145/3342197.3344534

This paper describes a user study in a driving simulator setup using OpenROUTS3D. Our major intention was analyzing the impact of different types and levels of latency on driving performance and the subjectively rated perceived workload of the participants. we conducted the user study on a multi-monitor setup with a steering wheel and pedals running OpenROUTS3D. Participants had to drive various scenarios on different maps with altered levels of latency. We chose maps and scenarios to follow typical driving situations. Our goal was combining scenarios that require exact handling and a overall precise control of the remote vehicle, e. g., topics that are likely to become more complex with introduced latency. We applied different levels of latency on the maps to measure its on the driving performance and subjective factors. Latency was either constant (RTT of 0 s, 0.3 s or 0.6 s) or varying. These values were based on previous findings and a literature review. In order to track the objective driving performance, participants' drive was logged in detail. Considered performance indicators were typical values such as lateral deviation, maximum steering angle, out of lane ratio, average speed and acceleration/deceleration. These objective metrics were supported by participants' subjective assessment of the perceived workload. The questionnaire for the subjective assessment consisted of self-defined questions combined with the NASA Task Load Index (NASA-TLX), typically used for assessing the subjectively perceived workload. The study was carried out at the Technische Hochschule Ingolstadt and 28 participants took part. Results indicated that the influence of latency on both factors, the objective measured driving performance and the perceived workload changed with increased latency. Based on the complexity of the scenario, different levels of latency might be acceptable. So, latency values up to 300 ms might be handled by trained operators. In comparison, we could not see a difference between fixed and varying latency. However, due to the scenario and map selection, these result need to be treated carefully.

My contribution on this paper was, besides writing the major part of the paper, the general idea. Together with the co-authors I designed the user study and guided the development of specific maps and scenarios that address the required driving situations to be investigated. The setup was also chosen by me and set up together with working students. The user study itself was supervised by a working student and me. The final analysis of the collected data and the related discussion was done together with the co-authors. I also mainly took care of addressing reviewers' comments.

ACM Author Rights

ACM exists to support the needs of the computing community. For over sixty years ACM has developed publications and publication policies to maximize the visibility, impact, and reach of the research it publishes to a global community of researchers, educators, students, and practitioners. ACM has achieved its high impact, high quality, widely-read portfolio of publications with:

- Affordably priced publications
- Liberal Author rights policies
- Wide-spread, perpetual access to ACM publications via a leading-edge technology platform
- Sustainability of the good work of ACM that benefits the profession

Choose

Authors have the option to choose the level of rights management they prefer. ACM offers three different options for authors to manage the publication rights to their work.

- Authors who want ACM to manage the rights and permissions associated with their work, which includes defending against improper use by third parties, can use ACM’s traditional copyright transfer agreement.
- Authors who prefer to retain copyright of their work can sign an exclusive licensing agreement, which gives ACM the right but not the obligation to defend the work against improper use by third parties.
- Authors who wish to retain all rights to their work can choose ACM's author-pays option, which allows for perpetual open access through the ACM Digital Library. Authors choosing the author-pays option can give ACM non-exclusive permission to publish, sign ACM's exclusive licensing agreement or sign ACM's traditional copyright transfer agreement. Those choosing to grant ACM a non-exclusive permission to publish may also choose to display a Creative Commons License on their works.

Post

Otherwise known as "Self-Archiving" or "Posting Rights", all ACM published authors of magazine articles, journal articles, and conference papers retain the right to post the pre-submitted (also known as "pre-prints"), submitted, accepted, and peer-reviewed versions of their work in any and all of the following sites:

- Author's Homepage
- Author's Institutional Repository
- Any Repository legally mandated by the agency or funder funding the research on which the work is based
- Any Non-Commercial Repository or Aggregation that does not duplicate ACM tables of contents. Non-Commercial Repositories are defined as Repositories owned by non-profit organizations that do not charge a fee to access deposited articles and that do not sell advertising or otherwise profit from serving scholarly articles.

For the avoidance of doubt, an example of a site ACM authors may post all versions of their work to, with the exception of the final published "Version of Record", is ArXiv. ACM does request authors, who post to ArXiv or other permitted sites, to also post the published version's Digital Object Identifier (DOI) alongside the pre-published version on these sites, so that easy access may be facilitated to the published "Version of Record" upon publication in the ACM Digital Library.

Examples of sites ACM authors may not post their work to are ResearchGate, Academia.edu, Mendeley, or Sci-Hub, as these sites are all either commercial or in some instances utilize predatory practices that violate copyright, which negatively impacts both ACM and ACM authors.

Distribute

Authors can post an Author-Izer link enabling free downloads of the Definitive Version of the work permanently maintained in the ACM Digital Library.

- On the Author's own Home Page or
- In the Author's Institutional Repository.

Reuse

Authors can reuse any portion of their own work in a new work of their own (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is notthe editor, requires permission and usually a republication fee.
- Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).
- Commercially produced course-packs that are sold to students require permission and possibly a fee.

Create

ACM's copyright and publishing license include the right to make Derivative Works or new versions. For example, translations are "Derivative Works." By copyright or license, ACM may have its publications translated. However, ACM Authors continue to hold perpetual rights to revise their own works without seeking permission from ACM.

Minor Revisions and Updates to works already published in the ACM Digital Library are welcomed with the approval of the appropriate Editor-in-Chief or Program Chair.

- If the revision is minor, i.e., less than 25% of new substantive material, then the work should still have ACM's publishing notice, DOI pointer to the Definitive Version, and be labeled a "Minor Revision of"
- If the revision is major, i.e., 25% or more of new substantive material, then ACM considers this a new work in which the author retains full copyright ownership (despite ACM's copyright or license in the original published article) and the author need only cite the work from which this new one is derived.

Retain

Authors retain all perpetual rights laid out in the ACM Author Rights and Publishing Policy, including, but not limited to:

- Sole ownership and control of third-party permissions to use for artistic images intended for exploitation in other contexts
- All patent and moral rights
- Ownership and control of third-party permissions to use of software published by ACM

Teleoperation

The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters

Stefan Neumeier
stefan.neumeier@thi.de
Technische Hochschule Ingolstadt
Ingolstadt, Germany

Philipp Wintersberger
philipp.wintersberger@carissma.eu
Technische Hochschule Ingolstadt
Ingolstadt, Germany

Anna-Katharina Frison
anna-katharina.frison@thi.de
Technische Hochschule Ingolstadt
Ingolstadt, Germany

Armin Becher
becherarmin@gmail.com
Technische Hochschule Ingolstadt
Ingolstadt, Germany

Christian Facchi
christian.facchi@thi.de
Technische Hochschule Ingolstadt
Ingolstadt, Germany

Andreas Riener
andreas.riener@thi.de
Technische Hochschule Ingolstadt
Ingolstadt, Germany

ABSTRACT

In the domain of automated driving, numerous (technological) problems were solved in recent years, but still many limitations are around that could eventually prevent the deployment of automated driving systems (ADS) beyond SAE level 3. A remote operating fallback authority might be a promising solution. In order for teleoperation to function reliably and universal, it will make use of existing infrastructure, such as cellular networks. Unfortunately, cellular networks might suffer from variable performance. In this work, we investigate the effects of latency on task performance and perceived workload for different driving scenarios. Results from a simulator study (N=28) suggest that latency has negative influence on driving performance and subjective factors and led to a decreased confidence in Teleoperated Driving during the study. A latency of about 300 ms already led to a deteriorated driving performance, whereas variable latency did not consequently deteriorate driving performance.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Networks**; • **Computer systems organization**;

KEYWORDS

latency, qos, teleoperated driving, user study

AutomotiveUI '19, September 21–25, 2019, Utrecht, Netherlands

ACM ISBN 978-1-4503-6884-1/19/09…$15.00
https://doi.org/10.1145/3342197.3344534

ACM Reference Format:
Stefan Neumeier, Philipp Wintersberger, Anna-Katharina Frison, Armin Becher, Christian Facchi, and Andreas Riener. 2019. Teleoperation: The Holy Grail to Solve Problems of Automated Driving? Sure, but Latency Matters. In *11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '19), September 21–25, 2019, Utrecht, Netherlands.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3342197.3344534

1 INTRODUCTION

Technological progress usually aims to change society to the better, and the list of potential advantages of automated driving technology is long. Automated vehicles (AVs) promise to reduce driver stress, parking costs, energy consumption, pollution, while increasing safety, productivity, mobility for non-drivers, or road capacity [21]. However, when looking on the streets today, we might realize that many of these advantages are for the long haul. Considering the SAE levels of automation [30], existing automated driving systems operate on level 2, and fully automated level 5 vehicles are not expected within the next years – actually, even when the technology is reliable, additional time will be needed for testing and regulatory approval [21]. In addition, recent incidents with AVs raised the question, if automation that requires a human driver as fallback authority can safely be implemented [11, 33]. A promising approach to solve such problems and bring AV technology to the customer earlier is Teleoperation. Teleoperated Driving (sometimes also referred to as "remote driving") is the remote control of a vehicle by a human in particular situations. For example, vendors of shared mobility could provide (level 3) vehicles that, for passengers, appear like fully automated vehicles, yet an off-board operator takes over if necessary. Another example are automated trucks, that may be able to drive alone on highways, but need support when reaching their operational design domain – in such scenarios, a small number of operators could potentially administer a larger number of vehicles. Further scenarios are software/hardware failures of highly autonomous vehicles [17]. Teleoperated Driving systems are already in use and are being developed further by different companies [6]. It is additionally required for example in California to allow driverless vehicle testing [7]. Thus, Teleoperated Driving is claimed to play an important role in the story of autonomous vehicles. Enabling Teleoperated Driving requires the utilization of wireless connections between the operator and the vehicle [27]. A technology that is already widely deployed has to be used to allow operation over a broad geographic area, and cellular networks could become the technology of choice, since they are able to meet required latency, bandwidth and packet loss demands [18]. Unfortunately, remote control will always go hand in hand with the problem of latency, emerging from the need to send/receive vehicle states and control signals from and to vehicles on the road. This paper tries to answers the question, how latency influences average human drivers in safely controlling a remote vehicle. To figure out the impact of latency, we conducted a driving simulator study, where participants had to drive different simulated tracks and deal with various levels of potentially changing latency. Initial investigations in this regard have already been published (e.g. [22]), yet a fine-grained evaluation concentrating on quantitative evaluation of driving performance measures has not been made.

2 RELATED WORK

A detailed investigation of about 150 papers regarding different aspects of human performance issues and mitigations for remote operations was done by Chen et al. [3], with the aim to point at areas where further research is required. Lichiardopol did a survey on teleoperation [20], presenting a general overview of teleoperation and covering important aspects such as telepresence and control issues. The basic concept of teleoperated robotics has been shown by Winfield in [35]. There, a teleoperated system consists of the three essential parts; remote device, teleoperation workspace and connection between both. Basic setups for Teleoperated Driving are shown by Neumeier et al. [27], Gnatzig et al. [13] and Shen et al. [32].

The ability to provide immersion, the telepresence, is a relevant prerequisite for Teleoperated Driving [2] and widely investigated. In [16] Hosseini and Lienkamp show that Teleoperated Driving could be improved by virtual-reality systems. Virtual-reality systems and multi-monitor setups are also compared by Georg et al. [12]. It is shown that head-mounted displays allow for a better immersion, but - in contrast to the aforementioned results - do not necessarily improve driving performance. Chucholowski [5] analyzed different display methods for teleoperated vehicles and showed that a predictive display can help to mitigate time delays. In [16] Hosseini and Lienkamp introduce a combined approach with virtual-reality and augmented sensor data to make driving safer.

To use Teleoperated Driving in a safe manner, the network is one of the most important factors. Network latency for Teleoperated Driving has been measured by Shen et al. [32], showing a average latency for video streaming of about 183 ms and 205 ms and an average latency of about 110 ms and 217 ms for vehicle controls in 4G and 3Gb respectively. It has been shown that driving a slalom course was possible even with 3G. Chucholowski et al. [4] faced an average latency of about 121 ms when submitting a video stream through 3G networks. Real-world measurements of cellular networks have been done by Neumeier et al. [28], where a median of Round-Trip-Time (RTT) of about 55 ms for UDP- and Ping-based measurements was seen. Kang et al. [17] set up a real-world testbed to analyze the behaviour of LTE and WiFi for Teleoperated Driving, showing that the median two-way latency was about 100 ms over the LTE network. One of the major considerations in Teleoperated Driving is the present existent latency, caused by the use of telecommunication networks. Luck et al. [24] investigated the effects of latency on remote operation and claim that constant latency is easier to deal with than variable one. This claim is confirmed by findings of Davies et al. [8] and Liu et al. [22].

User studies regarding Teleoperated Driving have been carried out by various research groups. In [22] Liu et al. conducted an user study with state-of-the-art LTE network performance and a small-scale vehicle. They claim that Teleoperated Driving over LTE does not work without supporting systems. Vozar and Tilbury [34] conducted an user study to explore the effects of latency. It is shown that the path-following score decreases with higher latency. A further, not Teleoperated Driving specific, user study was conducted by Nielsen et al. [29]. They introduced a combined 3D view and analyzed the results, showing that their approach improves the driving.

Although there is already a lot of research on Teleoperated Driving, including user-studies, most of it consists of single, small-scale scenarios. With the user study conducted in this paper, we overcome these issues by using more realistic, close to real-world scenarios combined with real-world measured latencies.

3 USER STUDY

We conducted an user study to identify the impact of latency on driving performance, workload, and user acceptance. The study took place in a static driving simulator with a multi-monitor setup as used by "Designated Driver" [7] and "Phantom Auto" [14]. Consequently, motion sickness potentially caused by artificial latency should be limited. We implemented different scenarios, along with different fixed and variable latencies, aiming to get close to real-world driving situations. The chosen scenarios reflect real-world driving, including parking and general vehicle handling. By evaluating both, behavioral (driving performance) and self-rating (e.g. NASA-TLX [15], self-defined questionnaire) data, we thereby want to answer the following research questions:

- **RQ1:** How do different constant latencies influence driving performance?
- **RQ2:** Does constant latency lead to better driving performance than variable latency?
- **RQ3:** What impact does latency have on subjective workload?

Setup and Scenarios

The experiment was conducted with an internally developed 3D driving simulator called *OpenROUTS3D* [25]. It allows for the creation of dynamic scenarios with different levels of input/output and inter-frame latency by applying buffers. Latency was configured for input and output to simulate both the latency of control commands, and the latency of a video stream, respectively. Latency was not displayed with its real value, but within groups indicated by colors and text. Groups were: no (0 ms), low (1 - 300 ms), medium (301 > 500 ms) and high (> 500 ms) latency, to match real-world Teleopearted Driving, where remote operators usually at least know the current level of latency to react appropriately. The physical driving simulator setup can be seen in Figure 1. Participants sat in the driving simulator with three 32" low-latency curved monitors (*AOC C32G1* with 1 ms Moving Picture Response Time (MPRT) [1]) arranged in a horizontal row. The chair was a *Playseat Evolution Alcantara* [31] gaming racing chair with *Logitech G29* steering and pedals [23] mounted to it. The driving simulator was separated from the investigators by movable partition walls. Thus, the participant was in his own private area, not observable by others during the study. However, the participant was able to talk to the investigators at any time if required. We abstained from using virtual reality head-mounted displays to avoid motion sickness during the study, which had a duration of up to 1 hour. Such a system could improve telepresence [16], but it does not necessarily lead to better driving performance, as more participant training with virtual reality systems might be required [12].

Considering latency as the main influencing factor, the driving simulators’s baseline end-to-end latency had to be known. Thus, the delay between steering wheel command and corresponding display output was measured. In order to do so, one of the steering wheel buttons was used to switch a rectangle on the center monitor from black to white. Using an external microcontroller with attached push button and photodiode allowed precise measurements of the time delays between button pressed and display output. A TSL250R photodiode was mounted on the front monitor within the

Figure 1: Setup of the driving simulator study.

color-changing rectangular. According to the datasheet, the sensor has an output rise-time of about 260 microseconds. As soon as the display output changes from black to white, a hardware interrupt on the microcontroller is triggered. The measurements revealed a median base latency (button pushed to display change) of about 66 ms, based on 50 measurements. It was also checked if the artificial input and output latency was applied correctly. Setting an artificial latency of 500 ms (250 ms each for input and output), the mean end-to-end latency was 567 ms, based on 21 measurements. This result was expected from the first measurements and thus latency was determined to be correctly applied. The framerate was set to constant 60 FPS. A more detailed explanation of the measurement setup in addition with the Unity-based source-code can be found as OpenSource-project on gitlab[1].

The user study consisted of five different scenarios. Every map was driven during daytime with sunny weather and did not contain any visible elements that are not driving-related. The streets had a width of about 7.0 m and thus followed the German "Richtlinien für die Anlage von Landstraßen (RAL)" [10]. The first map every participant drove was the practice map (see Figure 2). The green line indicates the center line of the track, whereas the red dot was the starting position of the drive. This map was driven without latency and ended either by user action (pressing button) or reaching the time-limit of 200 seconds. The scenarios for the experimental trials were `Parking`, `Snake`, `Pylon` and `LongTrack`. The most trivial scenario was the `Snake`, shown in second place in Figure 2. On this map the user had to stay on the street and follow it until reaching the final markers. This map contained some curves and was intended to check if it is possible to follow a simple street without complex steering maneuvers.

The `Pylon` map (third in Figure 2) consists of one big double curve with pylons placed on the center line of the track.

[1] https://gitlab.com/becheran/button-to-photon-latency-measure/

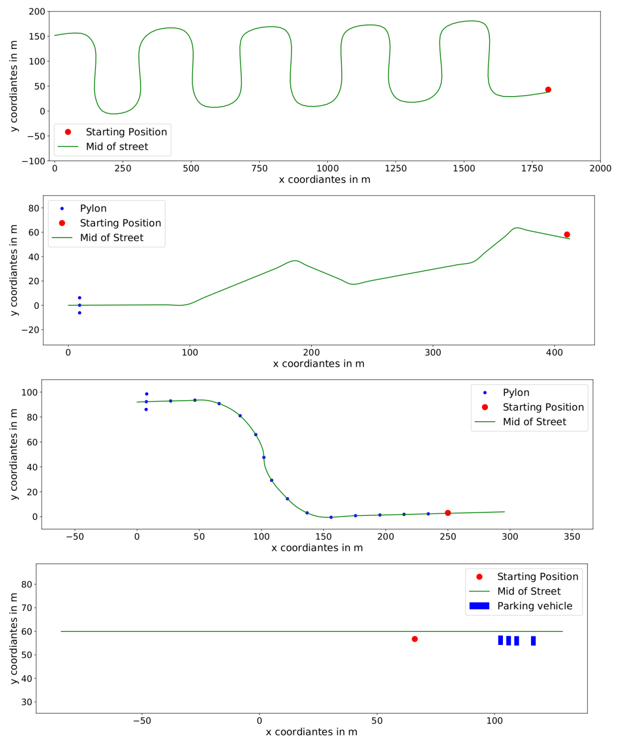


Figure 2: Maps of the user-study, including LongTrack/Practice, Snake, Pylon and Parking (from top).

Participants were instructed to drive in a zigzag pattern between the pylons. The curve was added to avoid acclimation of the driver as if he had only been driving on a straight street. This map was used to evaluate how participants behave in a handling course.

The Parking map (fourth in Figure 2) consists of four vehicles and a parking lot. Participants were instructed to park the vehicle inside this simulated parking lot and were allowed to decide if they preferred forward or backward parking (only with mirrors). The purpose of this map was to see if it is possible to safely execute a trivial everyday driving task, such as parking.

Finally, participants were required to drive the LongTrack scenario, which is also the first map in Figure 2. This track is the same as were introduced with the practice map, but with the difference that latency changed whilst driving was underway. The different areas, indicated with different colors, can be seen in Figure 3. Participants were instructed to follow the street, which was chosen to simulate a curvy rural road. The latency was changed to see how changing latency influences driving performance. This could happen frequently in real-world driving scenarios.

Every map (except the LongTrack map) was driven with three different levels of latency. The latency was either 0.0 s

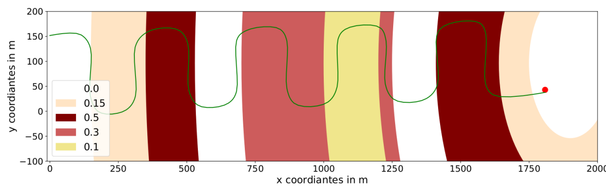


Figure 3: Latency areas for the LongTrack scenario.

(**no**), 0.15 s (**small**) or 0.3 s (**large**) each for input and output latency. The LongTrack had **varying** area-dependent delays between 0.0 s, 0.05 s, 0.075 s, 0.15 s and 0.25 s each for input and output latency. These delays were chosen based on the findings of Davis et al. [8] and Kang et al. [17], where a maximum acceptable latency of 700 ms [8] was identified. They also match with the results of real-world measurements conducted by Neumeier et al. [27]. Results of latency in video games as shown by Dick et al. [9] were also considered. Thus, latencies were selected to cover the range of previous findings, considering the constant presence of about 60 ms system latency.

Measurements

For a holistic contemplation of participants' teleoperated driving experience, we triangulated objective driving performance measures with participants' subjective self-ratings, which we collected by a questionnaire.

Driving Performance. To record driving performance, the current position of the pedals and steering wheel, the position of the vehicle and the current latency were logged along with additional information every frame (approximately every 16 ms). This log was analyzed with respect to the research questions and thus the following metrics were taken into account:

- **Lateral Deviation (MLP, SDLP)**: For position on the street, the Mean Lateral Position (MLP) and Standard Deviation of Lateral Position (SDLP) have been calculated [19] based on the center line. This values helped to indicate if driving was accurate or if there was a lot of deviation in the driving task. For the calculations, the vehicle's center point was used.
- **Max. Steering Angle**: This parameter describes the maximum angle participants steered to fulfill the driving task. This value (maximum of 35° in steering) helped to indicate how smooth a ride was and how participants were able to control the vehicle. The lower (except the Parking scenario) the value, the smoother a ride was (under the assumption that a large deviation from the track was compensated with sharp steering maneuvers).

- **Out of Lane Ratio**: This value represents the time (relative to the whole duration of a trial), in which a vehicle was not on the street. It was counted if at least one wheel crossed a side-lane marking. This value contributed to an understanding of the driving ability of the participant.
- **Avg. Speed**: Average speed indicated how fast participants were able to drive during a task.
- **Acceleration/Deceleration**: These values were calculated based on the speed-differences taken over time and only considered actions intended by the user. Thus, only accelerations and decelerations, which changed the vehicle's absolute speed at least (+/-) 2 km/h, were considered. They were assumed to be intended actions by the participants. Otherwise all rolling and speed maintenance actions would have been considered, but were not of interest to this evaluation. Both values could be used to determine the smoothness a drive had. The stronger the deceleration and acceleration, the bumpier the ride, thus, lower values are (potentially) better than higher ones.

Questionnaire. The questionnaire consisted of demographic information about age, gender, driving and video gaming experience to obtain some basic information on the participants (pre-driving questions). In addition, we assessed participants' trust in automated driving and public transport (*"I trust an automated vehicle/ public bus controlled by a computer/ by a teleoperated driver."*), as well as their confidence to remotely control a vehicle with, and without passengers on their own (*"I feel confident to be able to drive a vehicle with/without passengers teleoperated."*). To analyze the difference between expectations and reflection, we asked these questions before and after the drives (pre- and post-driving questions). To analyse the impact of the different latencies on participants' experience (post-latency questions), we utilized the NASA Task Load Index (NASA-TLX) [15] to quantify cognitive workload. Further, we implemented a self-defined questionnaire (in German, using a 5-point Likert scale). Thereby, we aimed to assess participants' stress (*"I perceived the situation during the trip as very stressful.", "I felt overwhelmed during the last ride.", Cronbachs* $\alpha > 0.8$), driving comfort (*"I perceived the situation during the trip as very comfortable."*), controllability (*"I had the vehicle well under control."*), and appropriateness of remote control of the experienced latency (*"I find the situation I have just experienced suitable for road traffic."*). Moreover, participants' were asked about the reasons of their problems regarding a specific latency.

Procedure

We utilized a within-subjects experimental design, where each participant completed each scenario multiple times, and with all combinations of scenario and latency, leading to a total of 10 trials (the three scenarios `Snake`, `Parking`, and `Pylon` with **no**, **small**, and **large** latency, as well as the `LongTrack` with **varying** latency) after the practice run. The order of conditions was randomized on two levels, each participant experienced a randomized order of latencies and within each latency, a randomized order of scenarios. At the beginning of the study, participants were introduced to the simulator environment and the basic concept of Teleoperated Driving. After answering the pre-driving questions, participants practiced driving on a test track for a maximum of 200 seconds.

Subsequently, the first of four relevant driving blocks started. A block consisted either of one fixed latency (**no**, **small** or **large**) with all maps (`Snake` and `Pylon` and `Parking`) except `LongTrack` in random order or the **variable** latency-based `LongTrack`. Thus, every participant drove the `LongTrack` once (if not considering practicing) and all other tracks three-times, once for each latency. The driving goals of each combination of map and latency were shown prior to every drive, e.g. follow the street, drive through the pylons, park between the cars. After each block, participants had to answer the post-latency questions. Subsequent to the final block, the questionnaire contained the additional post-driving questions. The procedure for each participant can be seen in Figure 4.

Participants

In total, 28 participants (6 female) took part in the driver study (M_{age}=26.89, SD=7.72). As we assume future teleoperated drivers to be young and tech-savvy, who might be, but not necessarily, interested in video gaming, we choose a diverse sample in the age between 10 and 30. All participants had a driver's license. Most of them drive more than 1,000 km per year (about 90%), while 2 participants drive more than 25,000 km per year. Self-reported driven distances ranges from 25% of participants with less than 5,000 km per year, 17% between 5,000 km and 10,000 km per year, 17% between 10,000 km and 15,000 km per year, 32% between 15,000 km and 25,000 km per year and 7% above 25,000 km per year. Participants' video game experience is diverse. It ranges from 35% who never play, to 40% who play once in a while, to only 25% who play frequently. Most played games are Shooter games (30%), followed by Strategy (26%), Jump and Run (12%), Simulators, Racing and Puzzle games (each 7%). Least played games are MMO and party games (each 5%).

4 RESULTS

Results are based on the different levels of latency and scenario, which accounts for the order throughout this section.

	Pylon Mdn (IQR)			Snake Mdn (IQR)			Parking Mdn (IQR)			Long Track Mdn (IQR)
Latency	**no**	**small**	**large**	**no**	**small**	**large**	**no**	**small**	**large**	**varying**
Lane Keeping Performance										
MLP	1.85 (.83)	2.13 (.55)	2.21 (.87)	2.52 (.73)	2.73 (.89)	2.66 (.52)	-	-	-	2.94 (.54)
SDLP	1.07 (.40)	1.27 (.45)	1.41 (.54)	.96 (.36)	1.07 (.58)	1.30 (1.02)	-	-	-	1.59 (.78)
Max. Steer Ang.	23.80 (8.10)	27.95 (8.48)	35.00 (6.07)	13.30 (4.55)	14.15 (5.90)	16.75 (13.92)	35.00 (6.2)	35.00 (5.65)	35.00 (8.10)	15.55 (7.48)
Out of Lane Ratio	0 (0)	0 (.03)	.03 (.08)	0 (.01)	0 (.03)	.02 (.08)	-	-	-	.027 (.05)
Speed and Acceleration/Deceleration										
Avg. Speed km/h	20.07 (5.49)	17.93 (8.60)	14.94 (6.18)	38.31 (13.92)	34.56 (13.38)	31.39 (9.66)	4.29 (3.31)	4.57 (4.46)	4.00 (3.03)	53.28 (3.31)
Mdn. Acc m/s^2	1.13 (.97)	1.15 (.97)	1.28 (.72)	1.23 (.62)	.96 (.90)	1.62 (1.11)	.80 (.36)	.85 (.66)	.89 (.40)	1.75 (.50)
Mdn. Decc m/s^2	3.05 (4.30)	2.84 (3.95)	2.14 (4.27)	1.53 (4.28)	5.82 (3.77)	4.38 (2.63)	1.36 (1.03)	1.76 (1.33)	1.64 (1.30)	4.99 (1.18)

Table 1: Medians and interquartile ranges of collected driving performance measures, showing the negative influence of increasing latency.

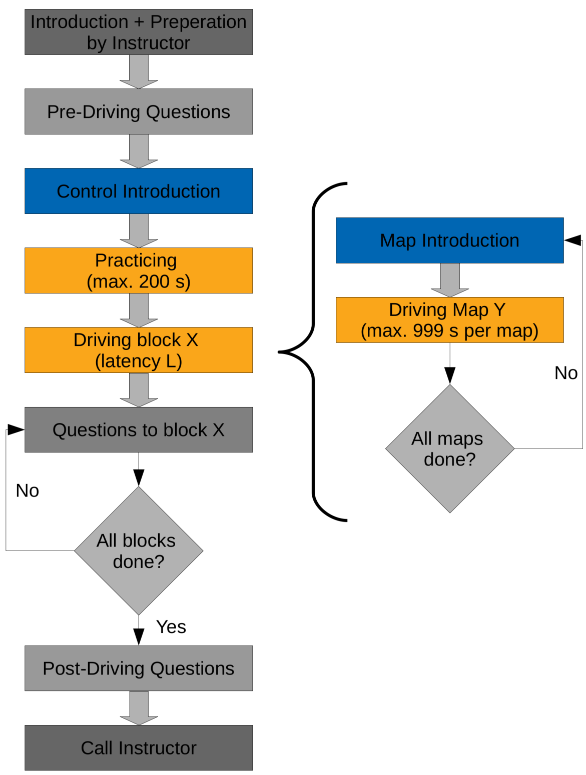


Figure 4: Procedure of the user study.

Driving Performance

To compare driving performance between the different latencies, we conducted multiple Friedman ANOVAs (data not normally distributed) within the given scenarios, since a direct comparison between the individual scenarios would not make much sense due to inherently different road configurations. However, to account for multiple tests given the 9 individual conditions, we adjusted the significance level accordingly to $.05/9 = .00\overline{5}$. In the following, we report the results of the statistical evaluation. An overview of descriptive statistics is depicted in Table 1.

Pylon Scenario. Considering the Mean Lateral Position (MLP), Friedman ANOVA indicated a significant difference between the three latencies ($\chi^2(2) = 12.07, p = .002$), whereas pairwise comparisons reveal differences between **no** and **small** ($p = .01$), and **no** and **large** latency ($p = .006$). However, both did not satisfy the alpha adjusted significance level. The Standard Deviation of Lane Position (SDLP) shows a similar picture, here Friedman ANOVA highlighted a significant effect too $\chi^2(2) = 11.14, p = .004$, and while the difference between **no** and **small** latency again was dismissed due to alpha correction ($p = .048$), we can accept the effect for the difference between **no** and **large** latency ($p = .004$). The Maximum Steering Angle for the Pylon scenario ($\chi^2(2) = 27.42, p < .001$), on the other hand, indicates significant differences between **no** and **small** ($p = .002$), as

well as **no** and **large** latency ($p < .001$). Similar for the Out of Lane Ratio ($\chi^2(2) = 19.70, p < .001$) – in the **large** latency condition, participants got out of the lane significantly more often, than in the lower latency conditions ($p = .001$).
Regarding driving speed and forces resulting from gas/brake pedal actuation, significant differences were found for the average speed ($\chi^2(2) = 30.50, p < .001$) between **no** and **small** ($p < .001$) and **no** and **large** ($p < .001$) latency, but no effects could be obtained for acceleration/deceleration values.

Snake Scenario. In the Snake scenario, no differences could be obtained for MLP, but Friedman ANOVA indicated a significant effect for SDLP ($\chi^2(2) = 13.07, p = .001$), that was significantly higher in **large**, compared to **no** latency ($p = .001$). Similar, the Maximum Steering Angle ($\chi^2(2) = 12.07, p = .002$) was significantly higher in the **large** than in the **no** latency condition ($p = .002$). No differences could be revealed for the Out of Lane Ratio.
Still, in this scenario various effects for speed and gas/brake pedal actuation were present. Considering speed ($\chi^2(2) = 33.50, p < .001$), significant differences were found between **no** and **large** ($p = .002$), as well as **small** and **large** ($p = .002$) latency. Also for acceleration ($\chi^2(2) = 12.50, p = .002$), a difference was present between **no** and **large** ($p = .003$) latency, but not for other pairwise comparisons. In contrast for deceleration, again both **small** ($p = .001$), and **large** ($p = .005$) latency differed significantly from the **no** latency condition.

Parking Scenario. Interestingly, in the `Parking` scenario no significant differences could be found. Here, we did not calculate lane keeping performance parameters (it was a very short scenario, where leaving the lane to drive into the parking spot was given by definition, resulting in the maximum steering angle in all cases), but also regarding speed, acceleration, and deceleration, no effects are present.

LongTrack Scenario. To quantitatively compare **varying** and fixed latency, we choose to evaluate only the Lane Keeping parameters MLP and SDLP in comparison to the `Snake` scenario. Other parameters would certainly differ due to the differences in the road configuration, and also MLP and SDLP must be considered with caution, but we still claim (especially SDLP) as a valid measure for comparison, since it is typically chosen to evaluated driving errors. For statistical evaluation we conducted a series of pairwise comparisons using Wilcoxon Signed Rank tests suitable for within-subjects evaluation.
Considering MLP, all investigated latencies **no** ($Z = -3.67 p < .001$), **small** ($Z = -2.48 p = .013$), and **large** ($Z = -2.66 p = .008$) were significantly smaller than the values obtained in the `LongTrack`, assuming an alpha-corrected significance level of ($.05/3 = .01\overline{6}$) to account for multiple comparisons.

	Pre TD Mdn (IQR)	**Post TD Mdn (IQR)**
With passengers	2.00 (2.00)	0.50 (1.00)
Without passengers	3.00 (2.00)	2.00 (2.00)

Table 2: Medians and interquartile ranges of participants confidence to operate an vehicle.

In contrast, the SDLP differed only significantly to the **no** latency condition ($Z = -4.01 p < .001$) – the **small** latency indicated a significant difference ($Z = -2.28 p = .023$), that however did not match the corrected alpha level.

Questionnaire

Pre- vs. Post-Driving. Regarding general questions before and after the experiment, a Wilcoxon signed-rank test (due to the nature of Likert scales) revealed that participants' trust in teleoperated vehicles (z=-2.99, p=.003) and public transport (z=-3.47, p = .001) significantly decreased after the experience of Teleoperated Driving (for both before: Mdn=2.0; for both after: Mdn=1.0). Further, participants' confidence to operate a teleoperated vehicle with (z=-4.11, p< .001) and without passengers (z=-3.12, p=.002) on their own significantly decreased as well. Thereby, we could identify a significant difference of feeling confident to operate a vehicle with or without passengers, when they were asked after experiencing Teleoperated Driving (z=3.90, p<.001), see Table 2.

Differences between Latencies (Self-Defined Questions). To compare the different latencies, we applied Friedman ANOVAs with Bonferroni correction for multiple comparison. Participants' overall stress significantly differs between all conditions ($\chi^2(3) = 26.83, p < .001$). Pairwise comparisons revealed that stress differentiates significantly between **no** and **large** (p <.001), between **no** and **varying** (p=.003), and between **small** and **large** latency (p<.023). Participants perceived comfort differs also between all conditions ($\chi^2(3) = 45.64, p < .001$). Pairwise comparisons showed a significant difference between **no** and **small** (p=.027), between **no** and **large** (p<.001), between **no** and **varying** (p<.001), and between **small** and **large** latency (p=.007). Perceived control differs also significantly ($\chi^2(3) = 47.03, p < .001$), pairwise comparison reveal differences between **no** and **small** (p=.016), **no** and **large** (p<.001), **no** and **varying** (p<.001), **small** and **large** latency (p=0.008). Thereby also the rated appropriateness of Teleoperated Driving on real road is significantly affected by different conditions ($\chi^2(3) = 33.86, p < .001$). However, a significant effect could only be identified

	No Mdn (IQR)	Small Mdn (IQR)	Large Mdn (IQR)	Varying Mdn (IQR)
Stress	0.50 (1.00)	1.00 (2.38)	2.75 (2.38)	2.00 (1.50)
Comfort	3.00 (2.00)	2.00 (2.00)	1.00 (2.00)	1.00 (2.00)
Control	3.00 (1.00)	2.00 (2.00)	1.00 (2.00)	1.50 (3.00)
Appropriateness	2.50 (1.00)	2.00 (1.00)	1.00 (1.00)	1.00 (2.00)

Table 3: Medians and interquartile ranges from the post-latency questions.

	No Mdn (IQR)	Small Mdn (IQR)	Large Mdn (IQR)	Varying Mdn (IQR)
Effort	1.00 (1.00)	2.00 (2.00)	3.00 (1.00)	2.00 (1.00)
Frustration	1.00 (1.00)	1.00 (3.00)	3.00 (1.00)	2.00 (3.00)
Performance	1.00 (2.00)	2.00 (2.00)	3.00 (3.00)	2.00 (1.00)
Mental Demand	1.00 (2.00)	1.00 (2.00)	3.00 (2.00)	3.00 (1.00)
Physical Demand	0.00 (1.00)	1.00 (2.00)	2.00 (3.00)	1.00 (2.00)
Temporal Demand	0.50 (1.00)	1.00 (2.00)	1.00 (2.00)	1.00 (2.00)
Overall Workload	4.50 (5.75)	7.50 (6.00)	13.00 (8.50)	10.00 (5.75)

Table 4: Medians and interquartile ranges from the Nasa-TLX questionnaire.

between **no** and **large** (p<.001), and **no** and **varying** latency (p=.001), see Table 3.

Differences between Latencies (NASA-TLX). Regarding participants' workload, results from the NASA-TLX (Table 4) reveals significant differences for the overall perceived workload (i.e., sum of all sub-scales), ($\chi^2(3) = 52.42, p < .001$). Thereby, pairwise comparisons reveal significant effect between **no** and **small** (p=.023), **no** and **large** (p<.001), **no** and **varying** latency (p<.001), and **small** and **large** latency (p<.001). Regarding sub-scales individually, significant effect could be revealed in all sub-scales: effort (E: $\chi^2(3) = 27.41, p < .001$), frustration (F: $\chi^2(3) = 31.17, p < .001$), performance (P: $\chi^2(3) = 30.71, p < .001$), mental demand (MD: $\chi^2(3) = 33.20, p < .001$), physical demand (PD: $\chi^2(3) = 25.88, p < .001$) and temporal demand (TD: $\chi^2(3) = 14.09, p = .003$). Thereby, pairwise comparison reveals significant differences between **no** and **large** (E: p<.001; F: p<.001; P<.001; MD: p<.001), and **no** and **varying** latency (E: p=.006; F: p=.010; P: p=.002, MD: p=.001) for the sub-scales effort, frustration, performance and mental demand. Participants' frustration significantly differs in addition between **small** and **large** (F: p=0.037), and **small** and **varying** latency (F: p=0.010), while mental workload differs additionally between **small** and **varying** latency (MD: p=.023). Physical demand only differs between **no** and **large** (PD: p<.001) latency. However, individual latencies do not affect the perceived temporal demand, no significant differences could be identified.

Problems of Teleoperated Driving. In the condition with **no** latency, 35.71% (n=10) of the participants reported to have no problems with Teleoperated Driving, 25.00% (n=7) complained about the clarity of the monitors, 14.29% (n=4) about the the control of the steering wheel, and 10.71% (n=3) about unpredictable delays during their input. Steering the pedals and lack of clarity of the driving route were only mentioned by respectively 7.14% (n=2).

Already with a **small** latency, only one participant reported to have no problems, however, 28.57% (n=8) complained about a constant delay during their input (not mentioned after no latency), 25% (n=7) complained about unpredictable delays, and respectively 17.86% (n=5) about controlling the steering wheel and clarity of the monitors. Only 7.14% (n=2) complained about the lack of clarity of the driving route.

After the condition with a **large** latency none of the participants reported no problems with Teleoperated Driving, or complained about the monitor clarity. However, here, constant delays (53.57%, n=15) and unpredictable delays (32.14%, n=9) were mentioned as major problems. Controlling the steering wheel (7.14%, n=2), pedals and lack of clarity of the route (respectively 3.57%, n=1) were mentioned only a few times.

While **varying** latency, 10.71% (n=3) reported to have no problems. The most mentioned problem is the unpredictability of the delays (60.71%, n=17). Controlling the steering wheel (10.71%, n=3), lack of clarity of the route and monitors (respectively 7.14%, n=2), and constant delays (3.57%, n=1) were mentioned only by a few participants.

5 DISCUSSION

In the following, we discuss the implications of our results on the proposed research questions. To get an overview on how a drive looked like, Figure 5 shows the latency-based results of a single participant on the `Pylon` track.

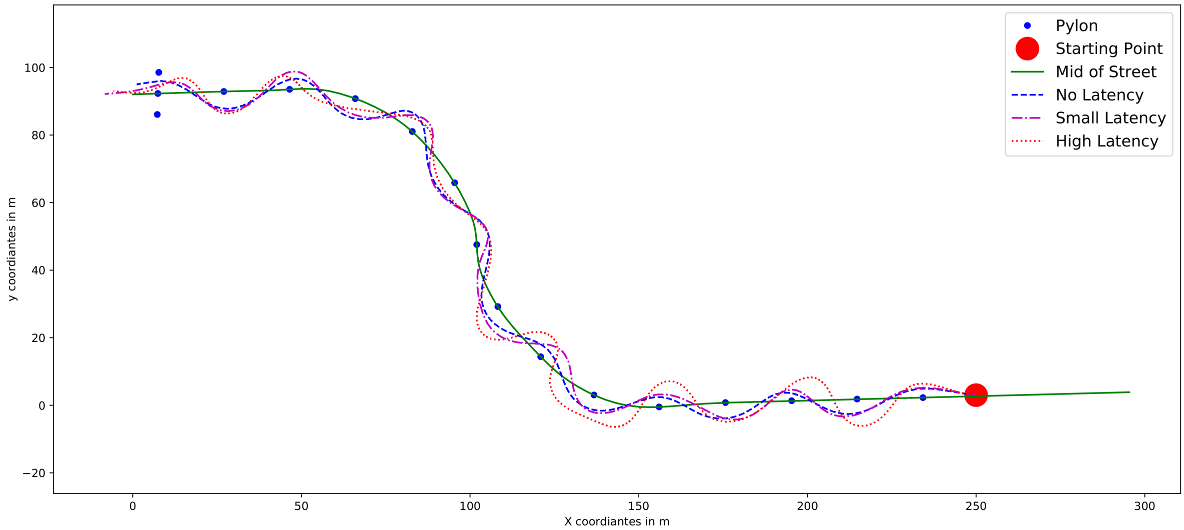


Figure 5: Driving results of one participant on the pylon scenario.

RQ1: Impact of Constant Latency on Driving Performance

The statistical evaluation of the different driving performance measures in the given scenarios reveals an interesting picture. While the **small** latency condition did not reveal a significant decrease for lane keeping (effects for MLP dismissed due to alpha-adjustment) performance, a significant difference could be obtained between **no** and **large** latency for SDLP in the `Pylon` and the `Snake` scenario. Since SDLP is a good predictor of lane keeping, this appears as driving with the **small** latency condition did not significantly worsen performance while **large** latency did. This might be supported by evaluation of the maximum steering wheel angle, that did not differ in the `Snake`, yet in the `Pylon` scenario between all investigated latencies. Also, participants significantly more often left the lane with **large** latency in the `Pylon` scenario (but no differences could be obtained for the **small** latency). In a similar fashion, speed and acceleration/deceleration differed only between **small** and **large**, as well as **no** and **large** latency in the `Snake` scenario. Overall consideration of these results makes us believe, that driving with **small** latency is acceptably good as with **no** latency, while **large** latency significantly worsens performance (comparable to the results of [34]). However, this has to be taken with caution - since due to the 9 conditions, the alpha-adjusted significance level was comparably low, and a higher number of participants might yield to additional significant effects. Still, we believe that the acceptable latency must be related to the actual driving scenario. In the `Parking` scenario, even no differences for whatever latency could be revealed, and most effects were visible in the `Pylon` scenario, that is somehow artificial compared to reality on roads (yet we believe it to be a good scenario for our investigations, and we see the high number of effects in this scenario as a confirmation).

Summarizing, driving performance decreases with increasing latency, but in very slow scenarios higher latencies might be acceptable, while in complex scenarios that require many fast adjustments (as in the `Pylon` scenario), even **small** latency could be too much. In scenarios with low complexity (such as the `Snake` scenario), low latencies could be acceptable. However, with other traffic objects on the road, even here **small** latency could become dangerous, but on the other hand, low latencies below 300 ms round trip time might be sufficiently handled by trained operators (study participants engaged in remote driving for the very first time). Still, further experiments are needed in typical everyday situations with unpredictable outcome.

RQ2: Fixed vs. Varying Latency

When comparing lane keeping performance between the `LongTrack` (**varying** latency) and the `Snake` scenario (other comparisons would not be applicable due to the inherently different types of driving tasks), the mean lateral position was significantly worse on the `LongTrack`, compared to all latencies in the `Snake` scenario. However, also here the results could emerge from the different road configuration. A more realistic assessment could be provided by investigation of the standard deviation of lane position (SDLP). Here, differences were only visible to the **small** latency condition, but again the road configuration could potentially have influenced the results. We can thus (at least in our scenario comparison, given the defined latencies) not confirm the results of Kang et al. [17] and others, who state that fixed latency leads to better driving performance than **varying**.

RQ3: Self-Rated Perception

Evaluation of subjective ratings for workload (NASA-TLX), as well as our self-defined items is only possible between the different latencies, as assessment was conducted only after experiencing all three different scenarios in a given latency. The overall workload of NASA-TLX is a good indicator to see how demanding different latencies were and if an average driver could deal with them. Results show, that significant differences in the overall workload are present between **no** and **small**, **no** and **large**, **no** and **varying** and **small** and **large** latency. For Teleoperated Driving this means, that besides the **large** and **varying** latency, even **small** latency leads to a significant difference in the overall workload and might already require trained operators. As there is a significance between **small** and **large** latency, the aim of every teleoperated system design should be a reduction of system latency, at least below the 300 ms, as trained operators might also struggle with higher latencies. This accompanies with the significant difference between **no** and **large** and **no** and **varying** latency, considering effort, frustration, performance and mental demand. Taking the self-defined questionnaire into account, participants felt significantly more stressed if driving with **large** or **varying** latency. The perception of comfort and control changed as latency was added artificially. A significant difference in the appropriateness of Teleoperated Driving in a specific scenario could be seen between **no** and **large** and **large** and **varying** latency, where **large** and **varying** latency are marked as inapplicable.

In general, the overall workload shows that there is already a difference between **no** and **small** latency. For applying Teleoperated Driving in real-world, a skilled and trained driver thus seems indispensable. This finding is assured by the multiselect answers participants gave, where latency (if present) was reported as a problem. From a system's perspective the latency should be kept constant and as low as possible, but at least below 300 ms.

6 LIMITATIONS AND FUTURE WORK

The study was carried out with a limited number of 28 participants. This can show a trend but is not sufficient to provide general conclusions. Participants were in a situation where no harm could be deployed by them or to them. Thus, it is likely that drivers would behave differently when driving vehicles in real-world and the possibility of harming others. Selected participants were not trained specific to Teleoperated Driving. Thus, they are not skilled in dealing with latency. Results with skilled and special-trained remote operators might differ. The user-interface was limited to the multi-monitor setup. Results might differ from the actual ones if utilizing a virtual-reality system setup [16]. The selected tracks and latencies only reflect an extract of real-world scenarios and did not consider other traffic participants. With latency the user-study only focused on one Quality of Service (QoS) aspect required for Teleoperated Driving. Besides the latency, bandwidth and packet loss are also important.

For future work the next step is to conduct a driver study with artificial traffic in real-world scenarios, e.g. driving through a city with artificial traffic and other QoS factors. The subsequent step is conducting real-world test drives. It will be further investigated if specially trained drivers are able to overcome latency and drive safely even with latency. This includes considerations if there are any specific attributes a remote operator should have. Finally, it has to be investigated which content is required for a remote driver and how this content can be displayed best, as some of the participants complained about varying latency even in scenarios where latency was constant.

7 CONCLUSION

We conducted an user study based on a virtual driving simulator to compare different levels of latency in Teleoperated Driving. The participants had to drive different typical real-world scenarios with either varying or constant latency. In addition to the objective driving performance, subjective workload and acceptance of teleoperation was assessed using questionnaires. Results show that latency has an influence on the driving performance. Even latency of about 300 ms led to a deteriorated driving performance. Variable latency did not consequently deteriorate driving performance. These objective results partly match with the subjective assessments of the participants, but large and variable latency lead to a higher overall workload. The confidence in Teleoperated Driving decreased during the user study as participants had to deal with constant/variable delay on their own. Thus it is important to consider all these results for real-world Teleoperated Driving. Mapping these results to real-world network measurements reveal, that today's cellular networks can provide required performance, but - as previously mentioned for example by [22] - further approaches (e.g. whitelisting [28], driver support systems and route planning [26]) are required. In general it can be said, that latency should be constant and as low as possible and only skilled and trained drivers should act as remote operators.

ACKNOWLEDGMENTS

This work is supported under the "Innovative Hochschule" program of the German Federal Ministry of Education and Research (BMBF) under Grant No. 03IHS109A (MenschIN-Bewegung), as well as the FH-Impuls program under Grant No. 13FH7I01IA (SAFIR). The authors would further like to thank Nino Bock for his support during the study.

REFERENCES

[1] AOC. 2019. C32G1 | AOC Monitors. https://eu.aoc.com/en/gaming-monitors/c32g1/specs. Accessed: 08.07.2019.

[2] Ryad Chellali and Khelifa Baizid. 2011. What Maps and What Displays for Remote Situation Awareness and ROV Localization?. In *Human Interface and the Management of Information. Interacting with Information*, Gavriel Salvendy and Michael J. Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 364–372.

[3] J. Y. C. Chen, E. C. Haas, and M. J. Barnes. 2007. Human Performance Issues and User Interface Design for Teleoperated Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37, 6 (Nov 2007), 1231–1245. https://doi.org/10.1109/TSMCC.2007.905819

[4] Frederic Chucholowski, Tito Tang, and Markus Lienkamp. 2014. Teleoperated Driving Robust and Secure Data Connections. *ATZelektronik worldwide* 9, 1 (01 Feb 2014), 42–45. https://doi.org/10.1365/s38314-014-0226-x

[5] Frederic Emanuel Chucholowski. 2016. Evaluation of Display Methods for Teleoperation of Road Vehicles. *Journal of Unmanned System Technology* 3, 3 (2016), 80–85.

[6] Alex Davies. 2017. Nissan's Path to Self-Driving Cars? Humans in Call Centers. https://www.wired.com/2017/01/nissans-self-driving-teleoperation/. Accessed on 21.10.2018.

[7] Alex Davies. 2019. The War to Remotely Control Self-Driving Cars Heats Up. https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars/. Accessed on 04.04.2019.

[8] J. Davis, C. Smyth, and K. McDowell. 2010. The Effects of Time Lag on Driving Performance and a Possible Mitigation. *IEEE Transactions on Robotics* 26, 3 (June 2010), 590–593. https://doi.org/10.1109/TRO.2010.2046695

[9] Matthias Dick, Oliver Wellnitz, and Lars Wolf. 2005. Analysis of Factors Affecting Players' Performance and Perception in Multiplayer Games. In *Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames '05)*. ACM, New York, NY, USA, 1–7. https://doi.org/10.1145/1103599.1103624

[10] Forschungsgesellschaft für Straßen- und Verkehrswesen, Arbeitsgruppe Straßenentwurf. 2012. Richtlinien für die Anlage von Landstraßen : RAL.

[11] Anna-Katharina Frison, Philipp Wintersberger, Andreas Riener, Clemens Schartmüller, Linda Ng Boyle, Erika Miller, and Klemens Weigl. 2019. In UX We Trust: Investigation of Aesthetics and Usability of Driver-Vehicle Interfaces and Their Impact on the Perception of Automated Driving. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 144, 13 pages. https://doi.org/10.1145/3290605.3300374

[12] J. Georg, J. Feiler, F. Diermeyer, and M. Lienkamp. 2018. Teleoperated Driving, a Key Technology for Automated Driving? Comparison of Actual Test Drives with a Head Mounted Display and Conventional Monitors*. In *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Maui, HI, USA, 3403–3408. https://doi.org/10.1109/ITSC.2018.8569408

[13] Sebastian Gnatzig, Frederic Chucholowski, Tito Tang, and Markus Lienkamp. 2013. A System Design for Teleoperated Road Vehicles.. In *ICINCO 2013 - Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*. Reykjavik; Iceland, 231–238.

[14] Mark Harris. 2018. CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads. https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads. Accessed on 28.11.2018.

[15] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139 – 183. https://doi.org/10.1016/S0166-4115(08)62386-9

[16] A. Hosseini and M. Lienkamp. 2016. Enhancing telepresence during the teleoperation of road vehicles using HMD-based mixed reality. In *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Gothenburg, Sweden, 1366–1373. https://doi.org/10.1109/IVS.2016.7535568

[17] Lei Kang, Wei Zhao, Bozhao Qi, and Suman Banerjee. 2018. Augmenting Self-Driving with Remote Control: Challenges and Directions. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications (HotMobile '18)*. ACM, New York, NY, USA, 19–24. https://doi.org/10.1145/3177102.3177104

[18] A. H. Khan, M. A. Qadeer, J. A. Ansari, and S. Waheed. 2009. 4G as a Next Generation Wireless Network. In *2009 International Conference on Future Computer and Communication*. IEEE, Kuala Lumpar, Malaysia, 334–338. https://doi.org/10.1109/ICFCC.2009.108

[19] G. Knappe, A. Keinath, K. Bengler, and C. Meinecke. 2007. Driving Simulators as an Evaluation Tool - Assessment of the Influence of the Field of View and Secondary Tasks on Lane Keeping and Steering Performance. In *20th International Technical Conference on the Enhanced Safety of Vehicles (ESV) (DOT-HS)*, Vol. 810-736. Lyon , France.

[20] S Lichiardopol. 2007. A survey on teleoperation. *Technische Universitat Eindhoven, DCT report* (2007).

[21] Todd Litman. 2019. *Autonomous Vehicle Implementation Predictions - Implications for Transport Planning*. Technical Report. Victoria Transport Policy Institute Victoria, Canada.

[22] Ruilin Liu, Daehan Kwak, Srinivas Devarakonda, Kostas Bekris, and Liviu Iftode. 2017. Investigating Remote Driving over the LTE Network. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '17)*. ACM, New York, NY, USA, 264–269. https://doi.org/10.1145/3122986.3123008

[23] Logitech. 2019. Logitech G920 & G29 Driving Force Steering Wheels & Pedals. https://www.logitechg.com/en-us/products/driving/driving-force-racing-wheel.html. Accessed: 08.07.2019.

[24] Jason P. Luck, Patricia L. McDermott, Laurel Allender, and Deborah C. Russell. 2006. An Investigation of Real World Control of Robotic Assets Under Communication Latency. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction (HRI '06)*. ACM, New York, NY, USA, 202–209. https://doi.org/10.1145/1121241.1121277

[25] Stefan Neumeier. 2019. OpenROUTS3D. https://git.io/fj8tn. Accessed: 16.05.2019.

[26] Stefan Neumeier and Christian Facchi. 2019. Towards a Driver Support System for Teleoperated Driving. In *22nd Intelligent Transportation Systems Conference (ITSC)*. IEEE, Auckland, New Zealand. Accepted Paper.

[27] Stefan Neumeier, Nicolas Gay, Clemens Dannheim, and Christian Facchi. 2018. On the Way to Autonomous Vehicles - Teleoperated Driving. In *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*. VDE, Dortmund, Germany, 1–6.

[28] Stefan Neumeier, Ermias Walelgne, Vaibhav Bajpai, Joerg Ott, and Christian Facchi. 2019. Measuring the Feasibility of Teleoperated Driving in Mobile Networks. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, Paris, France, 113–120.

[29] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks. 2007. Ecological Interfaces for Improving Mobile Robot Teleoperation. *IEEE Transactions on Robotics* 23, 5 (Oct 2007), 927–941. https://doi.org/10.1109/TRO.2007.907479

[30] On-Road Automated Driving (ORAD) committee. 2018. Taxonomy and Definitions for Terms Related to Driving Automation Systems for

On-Road Motor Vehicles.

[31] PlayseatStore. 2019. Playseat Evolution Alcantara - For all your racing needs. https://www.playseatstore.com/evolution-playstation.html#swatch_id=2156. Accessed: 08.07.2019.

[32] Xiaotong Shen, Zhuang Jie Chong, Scott Pendleton, Guo Ming James Fu, Baoxing Qin, Emilio Frazzoli, and Marcelo H. Ang. 2016. *Teleoperation of On-Road Vehicles via Immersive Telepresence Using Off-the-shelf Components*. Springer International Publishing, Cham, 1419–1433. https://doi.org/10.1007/978-3-319-08338-4_102

[33] Trent W Victor, Emma Tivesten, Pär Gustavsson, Joel Johansson, Fredrik Sangberg, and Mikael Ljung Aust. 2018. Automation expectation mismatch: Incorrect prediction despite eyes on threat and hands on wheel. *Human factors* 60, 8 (2018), 1095–1116.

[34] Steve Vozar and Dawn M. Tilbury. 2014. Driver Modeling for Teleoperation with Time Delay. *IFAC Proceedings Volumes* 47, 3 (2014), 3551 – 3556. https://doi.org/10.3182/20140824-6-ZA-1003.02275 19th IFAC World Congress.

[35] Alan Ft Winfield. 2000. Future Directions in Tele-operated Robotics. In *Telerobotic Applications*.

Publication [VI]: The Visual Quality of Teleoperated Driving Scenarios - How good is good enough?

[**VI**] Stefan Neumeier, Simon Stapf, and Christian Facchi. "The Visual Quality of Teleoperated Driving Scenarios - How good is good enough?" In: *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. Montreal, Canada: IEEE, Oct. 2020

The main focus of this paper is limiting the required bandwidth of video streams, that are typically required in teleoperated driving. This is a consequence of the limited bandwidth offered by cellular networks. In order to avoid complex pre-processing, the focus of this paper was utilizing already well known codec-based compression technologies to reduce the bandwidth requirements of a video stream. However, decreasing the required bandwidth was not the only factor to consider, as the video quality needs to stay above a certain quality level to enable remote drivers to sense the environment and spot potential dangerous situations. This ability also is influenced by the current driving situation and the prevailing environmental conditions. Thus, we conducted an online survey to identify suitable parameters. Therefore, a number of various video clips reflecting different driving scenarios was necessary, i. e., a selection of different environmental conditions and different street types. After we selected 10 suitable video clips, all of them were compressed with the following parameters: Constant Rate Factor (CRF), Resolution, Bitrate, Preset/Tune and Codec, which were altered. As this resulted in a large number of roughly 26000 different video clips that are too many to be shown to participants. Thus we had to reduce the number to be suitable for a user study. For each of the 10 video clips, we selected 5 different quality levels based on clustering the videos by their calculated Video Multi-Method Assessment Fusion (VMAF) scores. It could be seen that the highest influence on the required bandwidth has changing the parameters Bitrate, CRF and Resolution. Results of the survey indicated that bad weather and light conditions have a major influence on the usability of video clips in teleoperated driving. Rainy weather and bad light conditions were rated as not driveable even with the best presented quality. It also turned out that the utilized VMAF model could be used for a first assessment, but it requires an adjustment to the use-cases for teleoperated driving to allow for a better correlation between perceived quality and assessed quality. When mapping the obtained results to real-world bitrates, the range is between 288 kbps and 832 kbps per video clip, when ignoring the scenarios in which the quality was not rated as being sufficient.

The initial idea of this paper consisted of the thoughts I had regarding the compression algorithms used for teleoperated driving. With support of the co-authors I selected the suitable video-source and the driving scenarios. The calculation of the video clips, the development of the related software and the final selection of different qualities was done by me, involving the co-authors on the final decision. Together with the co-authors we designed and conducted the online survey. The final analysis and discussion of the overall results was done by me and frequently discussed with the co-authors. The major part of the writing was done by me in close exchange and with support of the co-authors. I also mainly took care of addressing reviewers' comments.

The Visual Quality of Teleoperated Driving Scenarios How good is good enough?

Conference Proceedings:
2020 International Symposium on Networks, Computers and Communications (ISNCC)
Author: [::Stefan::] [::Neumeier::]; Simon Stapf; Christian Facchi
Publisher: IEEE
Date: 20-22 Oct. 2020

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

The Visual Quality of Teleoperated Driving Scenarios How good is good enough?

Stefan Neumeier
Research Centre
Technische Hochschule Ingolstadt
Ingolstadt, Germany
stefan.neumeier@thi.de

Simon Stapf
Research Centre
Technische Hochschule Ingolstadt
Ingolstadt, Germany
simon.stapf@thi.de

Christian Facchi
Research Centre
Technische Hochschule Ingolstadt
Ingolstadt, Germany
christian.facchi@thi.de

***Abstract*—Teleoperated Driving can be a solution for situations that can not be solved by automated vehicles on their own. A human driver resolves the situation by controlling the vehicle remotely. Thus, drivers need to get an impression of the vehicle's environment. This is typically achieved by streaming video data through existing cellular networks. However, the available bitrate and the respective stream quality can vary drastically. This paper utilizes an user study to investigate, which video qualities are sufficient for different driving scenarios and how they are correlated with the required bitrates. Results show, that scenarios have a strong influence on the users' perceived quality ratings.**

***Index Terms*—teleoperated driving, video quality**

I. Introduction

Current development of future vehicles shows a distinct trend towards driver support and driver relief. This trend is set to continue and, via semi-autonomous vehicles, to produce fully automated vehicles in the future, enabling a "driver" to devote attention to non-driving relevant tasks. Finally, it is also possible that only passengers are inside the vehicle. Unfortunately, even highly automated vehicles can face dangerous situations and may be involved in serious crashes, e.g. as already happened with test vehicles [1]. In addition to crashes, automated vehicles can face further issues. Such issues might be raised by complex road-side works [2] or system failures such as confusion of the system or software and hardware malfunction as listed by Kang et al. [3]. Human interaction is required to solve such situations safely and quickly. This requires a suitable driver, who is inside the vehicle and able to take over control. For autonomous vehicles this is not guaranteed anymore. It is becoming more likely that vehicles are empty [4] or the humans inside are not able to take over control, e.g. in case of a medical emergency. Teleoperated Driving may be the solution in such situations. Teleoperated Driving is the remote control of a vehicle by a human in situations that require external support. Such systems are already used and developed by different companies and moreover also required by law in California when testing empty autonomous vehicles [5]. Thus, Teleoperated Driving is claimed to play an important role on the way to fully automated vehicles.

Enabling Teleoperated Driving requires the utilization of wireless connections, ideally – to achieve high coverage – with a technology that is already widely deployed. Cellular networks are therefore the means of choice. This already available technology is frequently improved and able to provide required demands regarding latency, bandwidth and packet loss [6]. However, cellular networks are subject to different influencing parameters and thus not able to provide constant latency, bandwidth and packet loss [7]. These values are highly relevant for Teloperated Driving and as such should never be below or above critical values, ensuring situational awareness and immersion, which are safety relevant. To provide immersion, most of current Teleoperated Driving systems, e.g. as in [2], use multiple cameras to provide the remote operator with environmental information.

The goal of this paper is to get an approximation of which video quality is sufficient for remotely controlling a vehicle under different driving scenarios and environmental conditions. A first estimate of minimum required bandwidths to achieve certain quality levels is given. To achieve these goals, an online user study with more than 100 participants was conducted. Video-clips of Waymo [8] were selected depending on the scenario, converted using various parameters of FFmpeg [9] and finally selected by their Video Multimethod Assessment Fusion (VMAF) [10] based quality rating.

II. Related Work

A general overview of Teleoperated Driving and control issues is presented in a survey of Lichiardopol in [11]. Following Winfield [12], the three essential parts, remote device, teleoperation workspace and connection between both define a teleoperated system. For this paper the connection is the decisive factor, being highly variable regarding bandwidth, latency and packet loss [7]. Liu et al. [13] utilized a LTE based user study with a small car to examine Teleoperated Driving. They claim that it does not work without assistant systems.

Telepresence, the eligibility to provide a feeling of the remote vehicle and its environment, is an important prerequisite for Teleoperated Driving [14] and thus already investigated. In [15] Hosseini and Lienkamp claim that virtual-reality can improve the performance of Teleoperated Driving. Georg et al. [16] were on the same track, comparing virtual-reality systems and multi-monitor setups. However, they show that virtual-reality systems do not necessarily improve driving

978-1-7281-5628-6/20/$31.00 ©2020 IEEE

performance, although they provide a better immersion. Chucholowski [17] analyzed different display methods for teleoperated vehicles and showed that a predictive display can help to mitigate time delays. Hosseini and Lienkamp [15] introduced a combined approach of virtual-reality and augmented sensor data in order to make remote driving safer.

In recent years, there has been remarkable progress in understanding the quality perception by the human visual system [18]. Vision can be seen as the dominant sensory organ. Studies show, that the overall perception is strongly dominated by vision. The human vision system needs a few seconds to adjust to changing quality levels [19]. Thus, the demand of controlled video quality adaptation is needed in order to reduce the negative effects of congestion on the stream, whilst providing the highest possible level of service and quality. In dark or blurred environments, sound or tactile perception can help capturing the situation [20]. However, this can be difficult to achieve for remote environments. These factors were all taken into account for preparatory work and survey.

Furthermore, evaluation methods must be considered. Subjective and Objective methods can be used to evaluate video quality. In the subjective quality assessment, the quality of the video material is judged by human viewers [21]. To obtain useful data, the Mean Opinion Score (MOS) [22], representing the perceived subjective quality, is an appropriate tool to be used. In order to achieve a general opinion of media quality – in this case video recordings – various methodologies have been developed and are used in the field of qualitative data in videos. Most of these tests have a MOS. MOS has a scale raging from 1 to 5 representing the categories: bad, poor, fair, good and excellent. Utilizing MOS to gather a qualitative feedback is a common strategy [23]. In their critical examination of the use of the MOS, Streijl et al. [22] state, that the MOS is often used before being considered as a suitable tool to explore the present research project. Nevertheless, MOS provides a suitable approach for this paper. The recommendation for the presentation of the stimuli "[...] range from 5 to 20 s in duration. Sequences of 8–10 s are highly recommended [...]" [24], were considered in this paper. A subjective evaluation can be considered as reliable approach to determine the quality of an image, as humans are usually the end users judging the quality. The subjective perception of participants and therefore the scale of quality may have a different meaning to each person.

In contrast to subjective results, objective metrics are calculated. Best-known techniques are: Peak signal-to-noise ratio (PSNR), Structural Similarity (SSIM) Index and the Multi-Scale SSIM (MS-SSIM). SSIM and MS-SSIM have proven to be a good predictor of image quality [25]. However, the SSIM especially has limitations for blurred and noisy videos [26] and all three metrics do not necessarily reflect the real perceived quality of viewers [10]. VMAF of Netflix is an improved approach that combines human vision modeling with machine learning [10], proving to provide better and more accurate results than the previous listed metrics. It was developed to accurately measure the perceived video-quality on a score ranging from 0 (worst) to 100 (best), as for Netflix's use cases the relation between (limited) bandwidth and video-quality matters.

III. Methodology

To get an idea of the suitability of different video quality levels in different driving scenarios, an online survey was conducted. In this user study, the participants were shown a total of ten videos and had to rate them based on video quality and usability for remote driving. Video rating was chosen, as no other influencing factors such as latency (already studied, e.g in [27]) should impact the perceived video-quality. Nevertheless, a few preparation steps were conducted to provide appropriate video-clips to the participants. Firstly, the video content was preselected. Subsequently, proper quality levels were chosen. Finally, the survey was conducted.

A. Preparation of the Video-Clips

Before the user study was started, it was vital to preselect appropriate video-clips and prepare them for the participants. Following Konstantopoulos et al. [28], suitable video-clips should reflect different weather and lighting situations.

1) Preselection of Dataset and Scenarios: The initial step consisted of finding proper sources to generate videos of. Due to legal obstacles and issues with blurring parts of the video, the authors could not record videos on their own and present them afterwards to the participants. It was therefore decided to utilize publicly available datasets such as KITTI [29], Lyft [30], Waymo [8], A2D2 [31] and Udacity [32]. Necessary selection criteria were image quality, scenarios and – as video-clips are generated – the image frequency. Offering the best overall package, the Waymo dataset was selected. It consists of 1,000 different driving segments, each with 20 seconds of length recorded at 10 Hz [33]. Even though typical Teleoperated Driving setups will have multiple cameras, only front camera images were used in this study as they are sufficient to rate the perceived quality and obtain an overview of the selected driving scene. Subsequently, the available data was ranked by visually analyzing it in terms of usability for the survey. Selection was made depending on *how exciting in the sense of complex a driving situation* was and on *how diverse the weather and lighting conditions* were. The first criteria was chosen to limit video-clips to dynamic driving scenarios including further traffic, lane crossing, etc. Many of the recordings consist of traffic jam or stopping at traffic lights and are, compared to a highway drive with a lot of vehicles, neither interesting nor challenging. The second criteria focuses on the fact, that different weather and light conditions may have an influence on the perceived quality. Finally, ten video-clips (Figure 1) were selected for the user study in order to achieve meaningful results already with about 50 participants.

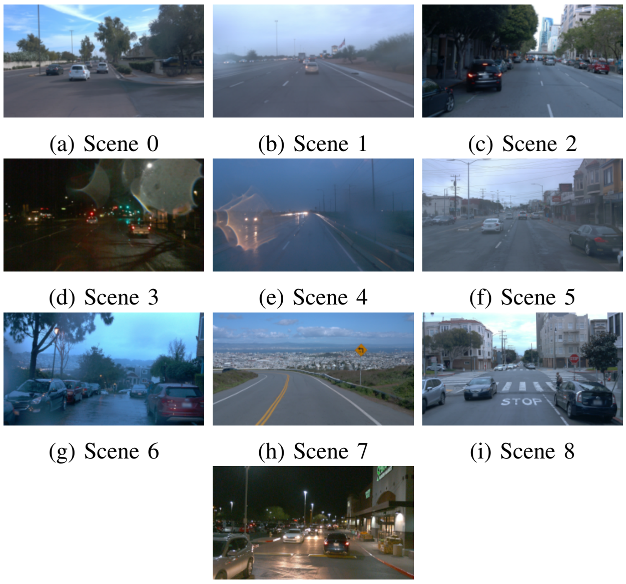
2) Calculation and Selection of Video Qualities: After selecting appropriate video-clips, they were cropped to reduce the original resolution (1920 x 1280) to a typical aspect ratio of 16:9 (Full-HD, 1920 x 1080; Cropping did not vanish any important objects). Subsequently, a practical framerate was estimated. Unfortunately, the images of the selected dataset were recorded at 10 Hz, which results in slow and inert video-clips. Different framerates in combination with different types of interpolation were tested in order to avoid this issue. However, after the videos were reviewed by five pre-study participants, it was decided to use a 20 Hz framerate without interpolation. Although the driving scenarios seem faster then, they are still usable with low influence on the perception as no driver interaction happens. Furthermore, this approach avoids the introduction of interpolation-based artifacts, e.g. flickering at patterned elements. Even though the A2D2 dataset was recorded at 30 Hz, the image quality is poor and the scenarios are less divers; the Waymo dataset remains the most appropriate option. Subsequently, different video qualities had to be generated. Different video qualities can be achieved through several configurations. For this paper, the parameters `preset`, `tune`, `bitrate`, `codec`, `crf` and `resolution` were altered, leading to a total of 25920 different video-clips.

CRF: The constant rate factor (CRF) defines the picture quality, but superposes the bitrate parameter if both are passed [34]. However, first results showed a slight difference in the resulting bitrate, if both parameters were set and thus this apparently "unnecessary" combination is also calculated. CRF was set to `24, 30, 36, 42, 48` and `None`. CRF values below 18 (H.264, libx264) are considered as visually lossless [34], and thus do not provide any contribution. Step-size was selected based on [34], where +6 is claimed to result in roughly half the bitrate. The resulting quality for the same CRF values is different in libx264 and libx265 (H.265). CRF 28 (libx265) visually corresponds to CRF 23 (libx264) [35].

Resolution: Another varied factor is the resolution. Resolutions altered were `1600x900, 1280x720, 960x540, 640x360, 480x270` and `320x180`. Full-HD resolution is used as input for Teleoperated Driving use cases, e.g. [2]. It served as ground-truth for VMAF calculations, but does not contribute to the survey and thus was ignored.

Bitrate: A specified bitrate acts as target bitrate during the encoding. Selected bitrates were `12400k, 6200k, 3200k, 1600k, 800k, 400k, 200k, 100k` and `None`. Bitrates were selected especially up to `3200k`, which the authors assume as reasonable real-world maximum for one video-stream. Higher values were included to consider them for analysis and are thus not the doubled values. They were chosen to stick closer to the lower bounds.

Preset/Tune: The presets consist of preconfigured options for the encoding. Slower presets achieve lower bitrates, but require longer encoding time. For latency-critical Teleoperated Driving, the encoding time is crucial. Therefore, the preset was set to either one of the two fastest presets (`ultrafast` or `superfast`). The tune parameter allows a further change of encoding parameters. `fastdecode` ("allows faster decoding by disabling certain filters" [34]) and `zerolatency` ("good for fast encoding and low-latency streaming" [34]) were used, as both are fittest for a latency-critical application such as Teleoperated Driving. [34]



(a) Scene 0 (b) Scene 1 (c) Scene 2
(d) Scene 3 (e) Scene 4 (f) Scene 5
(g) Scene 6 (h) Scene 7 (i) Scene 8
(j) Scene 9

Fig. 1: Scenario pool of the user study. (Source: [8])

Codec: The `H.264 (libx264)` [34] and the `H.265 (libx265)` [35] codec were chosen to reflect widely used and well known codecs. Even if there are more codecs, the selection fell to two widely used and widely supported codes.

Encoding happened on a Desktop-PC equipped with 16GB of RAM and an I7-7700 processor with 8 x 3.6 GHz (HT). Seven parallel threads were utilized, keeping one thread free for other tasks. The total computation time was about 10.45 days. The software versions used were: VMAF-Version (0.6.1), with Dev-Kit (1.3.1) and model *vmaf_v0.6.1.pkl* (1080p HDTV); FFmpeg (4.1.3) with libavcodec (58. 35.100); all running on a patched Ubuntu 19.04.

Based on the selected scenarios and the encoding options, all possible video combinations were generated using FFmpeg. Even though it is claimed that CRF will overrule bitrate, both were used simultaneously, since preliminary results indicated a difference when either both or only one value was set. Every video was further rated based on Netflix's VMAF. Not all of the resulting 25920 video-clips could be shown to the participants. Therefore, five quality levels for each video were selected, resulting in 50 video-clips for the survey. This selection happened by applying a clustering algorithms (sklearn version 0.20.3, *KMeans(n_clusters=5, random_state=0, n_init=25, max_iter=500)*) on the VMAF ratings. Finally, the video-clips with smallest distance to the cluster centers were selected for presentation. MS-SSIM and SSIM values were calculated but ignored, as they did not reflect the perceived video-quality of participants. These findings support the claim of Li et al. [10], of MS-SSIM and SSIM being not accurate enough, especially with noisy and complex video-clips like driving scenes. To avoid video-

	Parameters						Parameters				
Scenario	Type	#	Cluster	VMAF	Bitrate	Scenario	Type	#	Cluster	VMAF	Bitrate
0	Highway Sunny Day	1760	0	21.56	202.73	5	Suburban Rainy Day	1549	0	22.84	197.22
			1	37.01	401.56				1	39.73	392.68
			2	55.52	1580.36				2	58.90	1566.02
			3	73.53	3204.66				3	75.48	3174.58
			4	89.51	6224.72				4	89.61	6156.24
1	Highway Rainy Day	1650	0	23.16	189.61	6	Urban Rainy Day	1623	0	21.28	206.85
			1	36.77	402.28				1	36.61	402.08
			2	52.56	1588.01				2	54.37	1589.08
			3	69.81	3188.80				3	72.65	3228.56
			4	86.88	6295.02				4	89.20	6224.58
2	Suburban Sunny Day	1640	0	22.52	210.12	7	Rural Sunny Day	1551	0	22.60	207.23
			1	39.50	398.34				1	39.58	407.40
			2	59.08	1567.11				2	57.35	1584.98
			3	77.59	3272.64				3	73.99	3282.60
			4	91.92	6228.10				4	89.05	6397.42
3	Suburban Rainy Night	2038	0	22.62	237.50	8	Urban Sunny Day	1771	0	20.23	175.76
			1	36.87	414.80				1	33.13	346.72
			2	52.52	1631.63				2	48.45	1414.21
			3	69.17	3294.43				3	66.39	3016.14
			4	86.25	6397.35				4	88.21	5875.69
4	Highway Rainy Dawn/Dusk	2133	0	25.17	220.46	9	Urban Sunny Night	1891	0	22.28	177.00
			1	41.24	413.37				1	39.31	371.14
			2	56.46	1582.23				2	56.99	1555.78
			3	72.14	3168.80				3	73.77	3154.67
			4	87.98	6124.03				4	88.58	6133.62

TABLE I: Parameters of the selected scenarios for each of the groups (# of video-clips with VMAF above 15).

clips that were obviously too bad in quality, those with a VMAF rating of 15 (already below MOS' bad rating) and below were ignored. Subsequently, the results were visually inspected in a final round and claimed to be appropriate for the survey. Selected video-clips together with their scenarios and a short overview of important values can be seen in Table I. Although, there is a difference between the encoding speed of libx264 and libx265, it was neglected for selecting the videos of the survey. The authors argue, that the encoding speed is less important on the first hand if video-clips are only rated. For compatibility and comparability, all selected H.265 videos (34% of which were closest to cluster centers) were converted to H.264 losslessly for the presentation in the survey.

B. User Study

An online survey using SoSci Survey [36] was created to show the video-clips to participants and collect their ratings. In the first part of the survey, the participants were introduced to the survey (not to use handheld devices) and to Teleoperated Driving. This part was followed by questions about participants' demographic data. Subsequently, each participant was shown one randomly selected video-clip at a time. For each video-clip, the participant should rank the perceived video quality (MOS, `1 (Bad)` to `5 (Excellent)`). Further the participants should provide their opinion whether the displayed video quality would be sufficient for them to remotely control a vehicle (Options on a 4 point Likert-scale were: `yes, rather yes, rather no, no`). Participants were shown a total of ten randomly selected video clips. Thereby, the probability of biasing the participants by showing them only one quality level or only two specific scenarios was reduced. Participants were instructed to wait for buffering and watch the video-clips in fullscreen mode. They were told to avoid watching a video-clip multiple times, e.g. by manually reloading the page. The survey was designed to be completed in five to ten minutes and was not limited to a specific room or PC system. Thus, participants had to use their own devices. This was caused by the fact, that gathering different classes of population was easier this way. To prevent the unavoidable influence of improper setups in this open study, collected data included device resolution, deviceToPixelRatio, browser and operating system, enabling the authors to filter out monitors/setups that would distort the perceived video-quality. Addressing the perceived video-quality only, it was not necessary to conduct the study in a closed-loop manner, e.g. latency was not a factor for this study as it has been already investigated (e.g. by [27]). In order to guarantee variation, participants with different backgrounds, subject areas and ages were selected inside and outside the university by sending them impersonal invitation links.

C. First results on the calculated videos

The total size of all generated video-clips is about 34.62 GByte, with a median file size of about 335 KByte. The median VMAF value is about 32.50, ranging from 0 to 99.96. The reference video, which VMAF used as ground-truth for its rating, was the uncompressed Full-HD video-clip that was not shown to the participants. The median encoding speed is about factor 1.82, ranging from 0.1 to about 6. In addition to this general findings, detailed values are shown in Table II.

CRF: The CRF-influenced results behave as expected. The median bitrates decrease with a higher CRF. These bitrates also align with the VMAF results and the encoding speeds. For the encoding speed, a CRF of `24` has the median factor of 1.58,

	Bitrate								Tune		Preset	
	100k	200k	400k	800k	1600k	3200k	6200k	12400k	fast	zero	super	ultra
VMAF	10.96	24.06	39.09	52.57	65.25	75.73	80.78	83.66	30.97	34.63	30.87	34.18
Speed	1.75	1.86	1.73	1.54	1.28	1.28	1.24	1.0	1.83	1.81	1.76	1.88
Bitrate	102.01	200.09	398.91	796.83	1594.84	3189.11	6169.13	12327.82	243.66	296.32	253.47	294.19

	CRF						Resolution					
	24	30	36	42	48		320x180	480x270	640x360	960x540	1280x720	1600x900
VMAF	71.54	54.11	34.13	14.90	2.97		4.35	20.00	32.03	47.85	57.29	63.99
Speed	1,58	1.72	1.91	1.98	2.07		2.42	2.22	1.97	1.7	1.40	1.13
Bitrate	1298.33	529.67	214.84	91.33	45.82		79.09	133.72	200.92	346.50	508.91	731.95

TABLE II: Median results results of the encoded videos, split into the different parameters (Tune/Preset parameters are shortened). Speed indicates whether the encoding was faster (>1.0) or slower (<1.0) than realtime (1.0).

while a CRF of `48` has a median factor of 2.07. The factors in between are increasing step by step. The VMAF values indicate a decrease of the VMAF when increasing the CRF. The statement that changing the CRF by +/- 6 will double or bisect the bitrate [34], is roughly confirmed.

Resolution: The resolution has an impact on the bitrate. The higher the resolution, the higher the bitrate. The same applies to the encoding speed; the higher the resolution the lower the speed. Finally, and most interesting, the VMAF values are not increasing as expected. There is a steep jump between `320x180` to `480x270`, but other values align with increasing resolutions. Between `1280x720` and `1600x900` the difference is low.

Bitrate: Even though the bitrate parameter is not fully ignored when CRF is also set, it only has a small influence on the results (+/- 0.37% of the bitrate). Thus, only configurations where CRF was set to `None` are considered in the following when analyzing the bitrate. Otherwise results would not align to the bitrate parameter as one would expect and be biased too strong. The first thing that can be seen is, that the achieved bitrate aligns nicely with the configured bitrate. The VMAF values do not always follow a certain pattern with increasing bitrates, but a constant increase can be seen, even if the differences between `6200k` and `12400k` are rather small. The encoding speed does not consequently align with the increasing bitrates, but a trend can be seen, indicating that higher bitrates may lead to lower encoding speeds. Even if there are outliers (`200k`, `3200k`), the overall trend persists.

Preset/Tune: The influence of the tune, either `fastdecode` or `zerolatency`, is roughly comparable regarding the bitrates, even though `zerolatency` has an increased value of 296.32 KBit/s over 243.66 KBit/s. The encoding speeds are also comparable and the VMAF values align with the bitrates, indicating an around four point better rating for `zerolatency`. The results of the preset are comparable to the result of the tunes. The `ultrafast` preset provides a higher bitrate, a higher encoding speed and a higher VMAF rating compared to `superfast`.

Codec: One interesting aspect, especially for Teleoperated Driving, is the comparison of `H.265` and `H.264`. The median encoding speed of `H.264` is about factor 2.10, the one of `H.265` is factor 1.26. In pairwise comparison, `H.264` is also slightly faster than `H.265`. However, in 828 (3.2%, out of 25920) cases, `H.264` is slower than `H.265` and in 18 (0.07%) cases both have the same speed. Comparing the bitrates, `H.264` has a median bitrate of about 392.20 KBit/s, while `H.265` has a median bitrate of 191.02 KBit/s (48.70% of `H.264`). Nevertheless, in about 434 cases, `H.264` provides lower bitrates. The median VMAF values of `H.264` with 33.87 are slightly better than the median `H.265` values of 30.98; in 7896 cases, the VMAF values of `H.264` are better than `H.265`, in 171 cases they are the same.

IV. Results of the User Study

The survey period covered 28 days starting on January 2020. A total of 115 individuals were actively participating in the survey, e.g. they switched at least once pages. 95 out of these 115 (82%) participants finished the questionnaire properly and completely, preponderant with various setups. This number was further reduced by filtering out unsuitable setups, e.g. wrong aspect ratio or resolutions that are too low or too high might have a strong (negative) influence on the perceived video quality. Therefore, all resolutions (considering the deviceToPixelRatio and aspect ratio) below 1600x900 and above 2000x1250 were removed, leading to a total of 70 remaining participants (60%). Nevertheless, there were at least 10 ratings per video-clip. The average age of the participants was 30.41 years, including the rating of a participant who accidentally entered an age of 0. Excluding this outlier, the youngest participant was 21 years old, the oldest participant was 70 years. The distribution between male and female was 56 (80%) to 14 (20%), respectively. Most (68, 97%) of the participants had a driver's license, while one participant did not have a driver's licence. One participant was currently taking driving classes. The median time for a participant to finish the survey was 492 seconds. A median of 329 seconds was spent on loading, watching and rating the video-clips.

A. General Discussion of the results

The results of the survey can be seen in Table III, where they are split up into **Controlling** and **Rating**, indicating how participants would rate the perceived video quality and if they would trust themselves to control a vehicle remotely at this quality. Median values are used for rating the **Controlling** part, while average values are used for video quality ratings, as this is needed for the MOS. Performing the Kruskal-Wallis test for inter- and intra-scene analysis, it turned out that for **Controlling** and **Rating**, there is a significant difference using

S	Q	#	V	Controlling		Rating		S	#	V	Controlling		Rating	
				AVG	MED	AVG	MED				AVG	MED	AVG	MED
0	0	15	21.56	1.40	1.00	1.33	1.00	5	15	22.84	1.13	1.00	1.13	1.00
	1	16	37.01	2.38	2.00	2.38	2.50		14	39.73	1.64	1.50	1.64	2.00
	2	10	55.52	2.90	3.00	3.70	4.00		14	58.90	2.64	3.00	3.21	3.00
	3	17	73.53	3.00	3.00	3.53	4.00		14	75.48	3.14	3.00	3.86	4.00
	4	12	89.51	3.17	3.00	4.17	4.00		16	89.61	3.12	3.00	3.88	4.00
1	0	19	23.16	1.53	1.00	1.53	1.00	6	11	21.28	1.09	1.00	1.27	1.00
	1	14	36.77	1.50	1.00	1.57	1.50		11	36.61	2.18	2.00	2.91	3.00
	2	14	52.56	1.86	2.00	2.00	2.00		16	54.37	2.69	3.00	3.00	3.00
	3	15	69.81	2.60	3.00	3.07	3.00		13	72.65	2.77	3.00	3.23	3.00
	4	10	86.88	3.00	3.00	4.10	4.00		10	89.20	2.50	2.50	3.60	4.00
2	0	17	22.52	1.53	1.00	1.76	2.00	7	15	22.60	1.47	1.00	1.60	2.00
	1	17	39.50	2.12	2.00	2.59	2.00		15	39.58	2.27	2.00	2.00	2.00
	2	13	59.08	2.92	3.00	3.62	4.00		16	57.35	2.69	3.00	3.44	3.00
	3	15	77.59	3.00	3.00	3.53	4.00		11	73.99	2.73	3.00	3.82	4.00
	4	13	91.92	3.15	4.00	4.00	4.00		17	89.05	2.76	3.00	4.29	4.00
3	0	11	22.62	1.00	1.00	1.00	1.00	8	14	20.23	1.57	1.50	1.86	1.50
	1	17	36.87	1.35	1.00	1.47	1.00		13	33.13	2.00	2.00	2.08	2.00
	2	12	52.52	1.25	1.00	1.33	1.00		14	48.45	2.21	2.50	2.71	2.50
	3	17	69.17	1.53	1.00	1.76	2.00		14	66.39	2.64	3.00	3.36	3.00
	4	11	86.25	1.82	2.00	2.09	2.00		16	88.21	2.62	3.00	3.81	4.00
4	0	13	25.17	1.08	1.00	1.00	1.00	9	14	22.28	1.71	2.00	2.00	2.00
	1	16	41.24	1.06	1.00	1.06	1.00		15	39.31	2.27	2.00	2.27	2.00
	2	15	56.46	1.07	1.00	1.20	1.00		11	56.99	2.36	3.00	3.09	3.00
	3	12	72.14	1.42	1.00	1.58	2.00		13	73.77	2.92	3.00	3.31	4.00
	4	15	87.98	1.67	1.00	2.07	2.00		12	88.58	2.50	2.00	3.33	3.00

TABLE III: Results (**#** of ratings) of the survey, with green areas that indicate videos participants would rather trust as driver for Scenarios (**S**), Qualities (**Q**) and VMAF (**V**).

an α of 0.05. The pairwise Dunnett's test revealed a difference in the majority of the pairs (α of 0.05). However, there are cases, especially with a lower number of ratings, where no significant difference could be found. Nevertheless, this does not affect results of this paper too much, as tendencies can still be seen and first basic conclusions can be still drawn. One of the first results to be seen is that the average quality-ratings of the participants are only partly related to the results of VMAF ($r^2 = 0.56$). This correlation can be seen on Figure 2, where the average rated MOS was transferred to VMAF values (1.0 *MOS* is 20 *VMAF*; [10]). Lower VMAF values tend to be worse than actual user ratings, while at higher values things turn over. The average rated perceived qualities range from 1.00 (bad) for the quality level 0 to 4.29 (good) for the quality level 4. However, there are some outliers. There is the case, that the average rating does not reflect the order of the VMAF results. E.g. in scenario 0, the average rating of quality level 2 is rated higher than the one of quality level 3, while the VMAF results are opposite. The same happens in scenario 3 (Quality 1 and 2) and scenario 9 (Quality 3 and 4). However, the results are close and do not constitute a substantial difficulty.

More important are the results of the ratings of scenario 3 and scenario 4, as they do not reflect the results of the VMAF calculations at all. The ratings of the participants were indicating poor at best (2.09). Both scenarios have similar (bad) light and weather conditions, e.g. it is raining with little light. Under such environmental conditions, the used VMAF model does not predict the perceived video quality accurately. VMAF values were 86.25 and 87.98 at quality level 4 for scenario 3 and scenario 4, respectively, but user rating's of 2.09 and 2.07 (scaled to VMAF: 41.8 and 41.1) were poor.

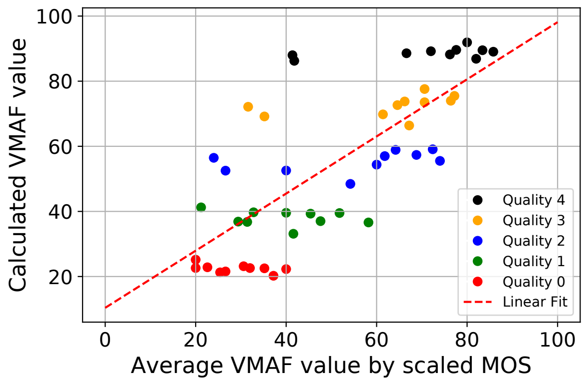


Fig. 2: Correlation ($r^2 = 0.56$) of calculated and rated VMAF values. MOS was scaled using the correlation of $1(MOS) == 20(VMAF)$.

B. Discussion based on Teleoperated Driving

The rating of the perceived video quality is related to the trust as driver in most cases. The MOS was at least 3.0 (fair) in cases, in which participants would trust themselves controlling a vehicle remotely. However, in scenario 6 and 9, ratings above 3.0 (both at quality level 4) were rated as insufficient for remotely controlling a vehicle themselves. Although, both of the quality levels received the highest MOS within the scenario, they were marked as not controllable. This has to be investigated further.

For most of the scenarios there is at least one quality level, where participants would rather accept to control a vehicle remotely. However, no quality level in scenario 3 or scenario

Scene	Minimal Bitrate	Scene	Minimal Bitrate
0	643.81	5	831.92
1	280.00	6	698.29
2	739.58	7	570.82
3	Undef.	8	687.23
4	Undef.	9	299.20

TABLE IV: Comparison of the different minimal VMAF metrics together with the minimal bitrate by keeping the encoding speed above real-time. Based on the ratings, it is not possible to obtain values for scenario 3 and 4.

4 was rated as sufficient for remotely controlling a vehicle. This results align with the poor ratings of the perceived video-quality. Another interesting fact is, that not always the highest quality level of 4 is considered as drivable. In the scenarios 6 and 9, the quality levels 2 and 3 were rated as acceptable, while quality level 4 was not. This findings strengthen the fact, that the used VMAF model needs to be retrained to be suitable for usage in real-world Teleoperated Driving scenarios. Unfortunately, a scenario-dependent threshold, stating that VMAF values above are suitable for driving a specific scenario remotely, can not be defined. Sometimes values above a specific threshold are usable, but higher values are not usable, e.g. in scenario 9.

To map the obtained results to real-world use cases, the minimal required bitrate for each of the scenarios will be presented. Quality levels are selected based on the lowest VMAF rating, which is considered (rated) sufficient for controlling a vehicle remotely. In addition, selected bitrates were required to match real-time encoding speed (encoding speed of at least 1). This allows for usage in real-world applications.

Ranging from 280.00 KBit/s to 831.92 KBit/s, results differ strongly, as can be seen in Table IV. Since no quality levels were rated as suitable for scenarios 3 and 4, no values can be specified. They might have to be above 3346.35 KBit/s (scenario 3) and 1044.44 KBit/s (scenario 4), requiring a quality superior to quality level 4.

However, these bitrates should be treated as a first approximation. They can help to identify tendencies and ranges of possible values and support the development of further systems, e.g. driver assistant systems or route planning algorithms. As a result, it can be said that for a typical Teleoperated Driving situation, the available bitrates of the network should be above the – scenario dependent – values mentioned in Table IV. Otherwise, further approaches such as removing colours or reducing the vehicle's speed as presented in [37] are required.

Summing up, it can be seen that the utilized VMAF model can partly be used as an indicator for the perceived video quality in Teleoperated Driving, but outliers exist. Especially bad weather combined with bad light conditions strongly influence the model. For most scenarios, participants rated at least one quality level as sufficient for remotely controlling a vehicle. However, for bad light and weather conditions, results were unsatisfying. The required minimal (assessable) bitrates ranging from 299.20 KBit/s to 831.92 KBit/s are scenario dependent and have to be adjusted based on the scenario. Considering the available bandwidth of the network as important factor, the two parameters bitrate (0.99, if CRF is not set) and CRF (0.75) showed the highest correlation (spearman) to the output bitrate and as such should be considered as the parameters to set. Resolution (0.45) also has an influence on the output bitrate, but its correlation is lower. Regarding speed, the highest correlation exist with the used resolution (0.62) and codec (0.53). Both factors have a major influence on the encoding speed. The investigation also indicates, that H.264 is (most of the times) faster and provides better ratings than H.265 and can be the codec of choice. For preset and tune, tested parameters are appropriate and have to be specified based on further factors, e.g. decoder type, etc.

V. Limitations

One of the main limitations of this paper includes the fact, that a certain (low) amount of participants rated a pre-selected variety of video-clips. With more participants and a wider variety of video-clips, the results might differ and be more significant, but the results are suitable to provide a first overview. Participants could watch a video-clip more than once, which offers them a different perception of a situation. However, this can be neglected in this paper as most of the participants (about 90%) did not watch a video-clip more than once. Another limitation of this work is the presentation of 20 Hz clips, that were recorded at 10 Hz. Thus, driving scenarios look faster than they actually are. Nevertheless, after testing different interpolation metrics, comparing different datasets and visually inspecting the results by multiple participants in a pre-study, it turned out, that the influence of this factor was considerably low as no driver actions resulting in specific feedback is present. The study was not conducted closed-loop, but with different setups/monitors. However, by filtering out unsuitable setups and investigating only the perceived video-quality, a closed-loop setup is not mandatory. The impact of latency on remote driving has been already investigated (e.g. in [27]) and is thus neglected in this study. The fact that only one camera-stream is shown could also be neglected, as video-clips were selected in a manner, that participants could get an overview of the scene and did not require additional information. This was confirmed by the participants filling free-text forms. Although the used VMAF model provided first results, it turned out the the model is not sufficient for driving scenarios under certain conditions, especially for rain and non-daylight conditions. However, for selecting videos and gaining first insights it was usable and more suitable than e.g. MS-SSIM.

VI. Conclusion and Future Work

The goal of this paper was to gain insights on which video quality is sufficient for Teleoperated Driving in specific situations. An online survey, in which participants had to rate different video-clips based on their quality, was carried out. With a total of 70 valid participants, it turned out that different environmental conditions require different video qualities. Bad light conditions, especially combined with rainy weather were overall rated as unsuitable and thus not mapped well by the used VMAF model. However, first values indicate, that under normal conditions, a bitrate from 299.20 KBit/s to 831.92 KBit/s could be sufficient for a single camera stream. Bitrate or CRF are the strongest influencing bandwidth parameters, while resolution and codec have the strongest influence on the encoding speed. Future work will be based on the results presented in this paper. As a first step, a new VMAF model has to be trained to reflect participants' ratings and be closer to the real-world perceived video quality. Subsequently, a larger survey with more participants and video-clips is required to obtain more significant results based on this new model. Further parameters for video encoding have to be tested then. This will include the use of different free codecs, such as VP8, VP9 and AV1. Furthermore, real-world measurements have to be conducted to validate the metric and develop a streaming system that fits the dynamic needs of Teleoperated Driving.

REFERENCES

[1] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in California," *PLOS ONE*, vol. 12, no. 9, pp. 1–20, 09 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0184952

[2] M. Harris, "CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads," https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads, Jan. 2018, accessed on 28.11.2018.

[3] L. Kang, W. Zhao, B. Qi, and S. Banerjee, "Augmenting self-driving with remote control: Challenges and directions," in *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, ser. HotMobile '18. New York, NY, USA: ACM, 2018, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/3177102.3177104

[4] A. Hars, *Self-Driving Cars: The Digital Transformation of Mobility*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 539–549. [Online]. Available: https://doi.org/10.1007/978-3-662-43782-7_57

[5] A. Davies, "The War to Remotely Control Self-Driving Cars Heats Up," https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars/, Mar. 2019, accessed on 04.04.2019.

[6] A. H. Khan, M. A. Qadeer, J. A. Ansari, and S. Waheed, "4G as a Next Generation Wireless Network," in *2009 International Conference on Future Computer and Communication*, April 2009, pp. 334–338.

[7] S. Neumeier, E. Walelgne, V. Bajpai, J. Ott, and C. Facchi, "Measuring the Feasibility of Teleoperated Driving in Mobile Networks," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, Paris, France, June 2019.

[8] "Waymo open dataset: An autonomous driving dataset," 2019.

[9] F. Developers, "FFmpeg," https://www.ffmpeg.org/, 2019, accessed on 20.12.2019.

[10] Z. Li *et al.*, "Toward A Practical Perceptual Video Quality Metric," https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, 06 2016, accessed on 18.12.2019.

[11] S. Lichiardopol, "A survey on teleoperation," *Technische Universitat Eindhoven, DCT report*, vol. 20, pp. 40–60, 2007.

[12] A. F. Winfield, "Future directions in tele-operated robotics," *Telerobotic applications*, pp. 147–163, 2000.

[13] R. Liu, D. Kwak, S. Devarakonda, K. Bekris, and L. Iftode, "Investigating remote driving over the lte network," in *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '17. New York, NY, USA: ACM, 2017, pp. 264–269. [Online]. Available: http://doi.acm.org/10.1145/3122986.3123008

[14] R. Chellali and K. Baizid, "What maps and what displays for remote situation awareness and rov localization?" in *Human Interface and the Management of Information. Interacting with Information*, G. Salvendy and M. J. Smith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 364–372.

[15] A. Hosseini and M. Lienkamp, "Enhancing telepresence during the teleoperation of road vehicles using HMD-based mixed reality," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 1366–1373.

[16] J. Georg, J. Feiler, F. Diermeyer, and M. Lienkamp, "Teleoperated driving, a key technology for automated driving? comparison of actual test drives with a head mounted display and conventional monitors*," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 3403–3408.

[17] F. E. Chucholowski, "Evaluation of Display Methods for Teleoperation of Road Vehicles," *Journal of Unmanned System Technology*, vol. 3, no. 3, pp. 80–85, 2016.

[18] Z. Wang and A. C. Bovik, "Modern image quality assessment," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, 2006.

[19] G. Ghinea, J. P. Thomas, and R. S. Fish, "Multimedia, network protocols and users—bridging the gap," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, 1999, pp. 473–476.

[20] G. F. Meyer and S. Wuerger, "Cross-modal integration of auditory and visual motion signals," *Neuroreport*, vol. 12, no. 11, pp. 2557–2560, 2001.

[21] I. T. UNION, "Subjective video quality assessment methods for multimedia applications," *SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS*, sep 1999.

[22] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[23] A. Ostaszewska and S. Żebrowska-Łucyk, "The method of increasing the accuracy of mean opinion score estimation in subjective quality evaluation," in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*. Springer, 2010, pp. 315–329.

[24] I. T. Union, "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of internet video and distribution quality television in any environment," *SERIES P: TERMINALS AND SUBJECTIVE AND OBJECTIVE ASSESSMENT METHODS*, mar 2016.

[25] A. Rehman and Z. Wang, "Reduced-reference ssim estimation," in *2010 IEEE International Conference on Image Processing*, Sep. 2010, pp. 289–292.

[26] C. Li and A. C. Bovik, "Content-partitioned structural similarity index for image quality assessment," *Signal Processing: Image Communication*, vol. 25, no. 7, pp. 517–526, 2010.

[27] S. Neumeier, P. Wintersberger, A.-K. Frison, A. Becher, C. Facchi, and A. Riener, "Teleoperation: The holy grail to solve problems of automated driving? sure, but latency matters," in *Proceedings of the 11th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '19. Utrecht, Netherlands: ACM, 2019. [Online]. Available: https://doi.org/10.1145/3342197.3344534

[28] P. Konstantopoulos, P. Chapman, and D. Crundall, "Driver's visual attention as a function of driving experience and visibility. using a driving simulator to explore drivers' eye movements in day, night and rain driving," *Accident Analysis & Prevention*, vol. 42, no. 3, pp. 827–834, 2010.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[30] R. Kesten *et al.*, "Lyft level 5 av dataset 2019," https://level5.lyft.com/dataset/, 2019.

[31] Geyer, Jakob and others, "A2d2: Aev autonomous driving dataset," http://www.a2d2.audi, 2019.

[32] Udacity, "GitHub - udacity/self-driving-car: The Udacity open source self-driving car project," https://github.com/udacity/self-driving-car, accessed on 08.01.2020.

[33] W. LLC, "About – Waymo," https://waymo.com/open/about/, accessed on 08.01.2020.

[34] FFmpeg Developers, "Encode/H.264 - FFmpeg," https://trac.ffmpeg.org/wiki/Encode/H.264, accessed on 22.01.2020.

[35] ——, "Encode/H.265 - FFmpeg," https://trac.ffmpeg.org/wiki/Encode/H.265, accessed on 22.01.2020.

[36] D. J. Leiner, "Our research's breadth lives on convenience samples a case study of the online respondent pool "sosci panel"," *SCM Studies in Communication and Media*, vol. 5, no. 4, pp. 367–396, 2017.

[37] S. Neumeier and C. Facchi, "Towards a Driver Support System for Teleoperated Driving," in *22nd Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct 2019.

Publication [I]: Data Rate Reduction for Video Streams in Teleoperated Driving

[**I**] Stefan Neumeier, Vaibhav Bajpai, Marion Neumeier, Christian Facchi, and Jörg Ott. "Data Rate Reduction for Video Streams in Teleoperated Driving". In: *IEEE Transactions on Intelligent Transportation Systems* 23.10 (Oct. 2022), pp. 19145–19160. ISSN: 1524-9050. DOI: 10.1109/TITS.2022.3171718

The main contribution of this paper is based on the idea, that the overall achieved bandwidth reductions of the work "The Visual Quality of Teleoperated Driving Scenarios - How good is good enough?" [**VI**] can be enhanced if smarter pre-processing approaches are used. Therefore, our first goal of this work was to reduce the bandwidth trough several steps. At first we investigated the use of different codec parameters that influence the visual quality, but may help to reduce the bandwidth consumption. Utilizing those evaluated parameters, we introduced the general idea, of how to split a camera-stream into important objects and the remainder. This allows the remainder to be blurred by a bilateral filter removing details but keeping surfaces. Important objects are built up step by step. Initially we considered the lane in front of the vehicle, but further enhanced it with the use of object detection based on two Machine Learning (ML) approaches. We either used the MobileNet or EfficientDet model to account for a very fast or very accurate model detecting vehicles, pedestrians, etc. Finally, the blurring approach was taken to the extreme by blurring based on a field of view. Results showed, that the average bandwidth improvement can be up to 467 kbps, which is about $1/3$ of the original required bandwidth. However, not all of the applied compression methods may be useful for teleoperated driving. Therefore, we conducted a user study in which participants had to evaluate the Mean Opinion Score (MOS) and the driveability for the different parameter combinations. It turned out, at least one set of combinations per scenario was rated driveable. The results indicate that an average improvement of about 247 kbps could be achieved across all video clips rated driveable. Based on the fact that different combinations may be best suited for different environmental conditions, a proposal system was introduced. This system was based on user study results and helps to always suggest the best suited combination of parameters.

The initial idea to this paper was developed by me when thinking of further reducing the bandwidth requirements for the video-stream from the vehicle to the remote operator. Together with the co-authors I selected suitable approaches to achieve the bandwidth reduction. The software to produce the videoclips based on the selected approaches was selected and developed by me. The analysis of the data and the discussion of the results was done by me, but always with support and in close exchange with the co-authors. The online survey was set up, conducted and analyzed mainly be me, being in close exchange with the co-authors. The proposal system was designed by me based on discussion with the co-authors. The major part of the writing was done by me, but took place in close exchange and with support of the co-authors. I also mainly took care of addressing reviewers' comments.

Data Rate Reduction for Video Streams in Teleoperated Driving

Author: Stefan Neumeier

Publication: IEEE Transactions on Intelligent Transportation Systems

Publisher: IEEE

Date: October 2022

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Data Rate Reduction for Video Streams in Teleoperated Driving

Stefan Neumeier, Vaibhav Bajpai, Marion Neumeier, Christian Facchi and Joerg Ott

***Abstract*—With the pioneering introduction of autonomous vehicles, system failures while driving from A to B are more likely to occur. In such scenarios one option is to hand back the control to the human driver, if someone suitable is inside the vehicle. Teleoperated Driving, the remote control of vehicles by human operators, can be a solution to scenarios without suitable drivers inside. A video stream is used to provide operators with an overview of the vehicle's environment and support for a safe remote control. By utilizing cellular networks as wireless communication medium for Teleoperated Driving, the available bandwidth is a limiting factor. This paper introduces a multi-step approach to lower the bandwidth requirements, which is achieved by initially splitting the single video stream into two parts: One part conveying the original video information restricted to important objects and the remainder, to which various filters are applied. Results show that this approach can lead to a decreased bandwidth consumption. These results are validated with a user study, where participants had to rate the perceived video quality and the driveability for the different combinations. This user study shows that, for every investigated scenario, at least one combination of parameters (applied filters) was rated driveable. Finally, the results are used to sketch a system that infers specific combinations of parameters based on the environmental conditions and the available bitrate.**

***Index Terms*—bandwidth optimization, teleoperated driving, user study, video stream**

I. Introduction

AUTOMATED vehicles promise to reduce driver stress, parking costs, energy consumption and pollution, while increasing safety, productivity, mobility for non-drivers and road capacity [1]. However, when assessing the situation on streets, it becomes apparent that many of these advantages are for the long haul. Considering the SAE levels of automation [2], existing purchasable automated driving systems operate on level 2, and fully automated level 5 vehicles are not expected within the next years – even if the technology is reliable, additional time will be needed for testing and regulatory approval [1]. In addition, recent incidents with automated vehicles raised the question, if automation that requires a human driver as a fallback authority can safely be implemented [3], [4]. A promising approach to solve problems of automated vehicles and bring such technology earlier to the customer is Teleoperated Driving. Teleoperated Driving is the remote control of a vehicle by a human operator in situations, where autonomous vehicles reach their system borders and have no suitable driver aboard. Possible scenarios are software and hardware failures on highly autonomous vehicles [5] or situations that may not be solved autonomously by highly automated vehicles, e. g. complex road-side works [6] or valet parking [7] in crowded and complex inner-city areas. This is when Teleoperated Driving comes into play, as human operators can contribute with their skills and knowledge. Teleoperated Driving systems are already being developed by different start-ups such as StarSky Robotics, Phantom Auto, Designated Driver, huge car manufacturers like Nissan [8] and telecommunication companies like Ericsson [9]. Furthermore, for testing driverless vehicles in the State of California (US), the ability to teleoperate is required by law [10]. To enable Teleoperated Driving in large geographical areas, wireless communication technologies need to be utilized [11]. In particular, cellular networks – especially modern standards such as LTE and 5G – are widely deployed and can provide the required demands regarding latency, bandwidth and packet loss [12]. However, despite the continuous evolution of cellular technologies, those networks still suffer from latency- and bandwidth-related issues. It is important, that these barriers are overcome, aiming to allow a safe use of Teleoperated Driving, i. e., the operator can perceive the environment and provide appropriate steering commands in time.

One of the main barriers is the ability of the teleoperator to perceive the vehicle's environment, which is usually achieved by providing a video stream of the environment. Yet, live video streams require large bandwidths and therefore can prohibit Teleoperated Driving in areas with low bandwidth provided by cellular networks.

This paper addresses the issues of bandwidth requirements by answering the research question: *How to reduce the bandwidth requirements of video streams in Teleoperated Driving*.

To this end, this paper provides three major contributions.
1.) Transformations of the video stream to require less bandwidth and allow the utilization of Teleoperated Driving in a larger geographical area, e. g. splitting up the stream into two separate parts for important objects and the remainder, applying different filters and putting it back together into one stream before transmission are investigated.
2.) To validate the findings with respect to the usability in

Manuscript received August 17, 2021; revised February 21, 2022; accepted April 12, 2022. The Associate Editor for this article was H. Huang. (*Corresponding author: Stefan Neumeier.*)

Stefan Neumeier is with C-ECOS at Technische Hochschule Ingolstadt and with the chair of Connected Mobility at Technical University of Munich, Ingolstadt/Munich, Germany (stefan.neumeier@thi.de).
Vaibhav Bajpai is with the CISPA Helmholtz Center for Information Security, Hannover, Germany (bajpai@cispa.de).
Marion Neumeier is with C-IAD at the Technische Hochschule Ingolstadt, Ingolstadt, Germany (marion.neumeier@carissma.eu).
Christian Facchi is with C-ECOS at the Technische Hochschule Ingolstadt, Ingolstadt, Germany (christian.facchi@thi.de).
Joerg Ott is Chair of Connected Mobility at Technical University of Munich, Munich, Germany, (ott@in.tum.de).

Digital Object Identifier 10.1109/TITS.2022.3171718

real-world scenarios, a user study in which participants have to evaluate the driveability and the perceived video-quality for the modifications as introduced by contribution 1 is conducted. **3.)** A system design that considers the previous results in order to propose the best suitable video modifications based on the available bitrate and the environmental conditions is outlined.

Therefore, this paper investigates different approaches by means of extensive experimentation, measurements and a user study. Algorithmic synthesis and the integration with a congestion control algorithm are not scope of this work. The integration of such an approach in a typical multi-monitor setup [6], the re-creation of a 3D picture or the utilization of additional sensors are also not part of this work.

This paper is organized as follows. Section II discusses related work and indicates the need for an innovative and new approach. In Section III the applied methodologies and the dataset together with the results of the experiments are addressed. Subsequently, Section IV presents the user study and discusses the obtained results, while Section V describes an inherited potential system designed considering previous results. The limitations of this work are shown in Section VI. Finally, Section VII concludes the paper and provides an outlook on future work.

II. Related Work and Background

Teleoperated systems are already used in various fields nowadays. Through the wide range of operations, diverse strategies and technologies are needed. One example of teleoperated systems are *Mars Rovers*, which are independent devices on Mars that are controlled from the Earth by submitting commands for time-delayed actions that are executed by the rover in its environment [13]. Another example are *Unmanned Aerial Vehicles (UAVs)* that are controlled remotely but also able to handle specific tasks autonomously [14].

Teleoperated systems, e. g. UAVs as in [14], usually consist of the three main parts (following [15]): Teleoperated device (robot), Teleoperation workspace and Communication link, which is the communication between devices and workspaces. The teleoperated device is a remote device. Its hard- and software mainly depends on the intended usage scenario. Commonly, a device is equipped with sensors, providing an environment's sense to the operator. In most cases, this sensor is a camera system, but also other sensors such as LiDAR [16] can be involved. The teleoperated device is additionally equipped with hardware to transmit and receive data and commands. Furthermore, hardware to execute the received commands is required. Distant from the teleoperated device, there is an interface for the operator in the workspace. This interface displays sensor data from the remote device. Additionally, the workspace enables the operator to control the remote device by providing (sequential) commands. For exchanging data and steering commands between the operator's workspace and the remote vehicle, a wired or wireless connection is required.

A major problem in remotely controlling a vehicle is the connection's quality of service, e. g. bandwidth, latency and reliability between the teleoperator and the remote vehicle. With LTE-Advanced, the uplink rate is increased up to 1.5 Gbps [17], which should be enough for transmitting the required video streams and control commands. Unfortunately, mobile connections suffer from potential high delays and packet loss [18]. Further, the data rate can drop drastically depending on the mobile cell workload. 5G could mitigate these problems, but future measurements under real-world conditions need to prove such claims. In addition to data compression, current approaches employ lightweight protocols like UDP in order to reduce communication overhead [19] and decrease the required bandwidth. UDP helps to avoid retransmission and head-of-line blocking and, hence, can help to drastically reduce the latency.

Research has shown several approaches to help mitigate the impediments of Teleoperated Driving induced by the required connection quality. The main goal is to assist the operator so that he has the impression of physically sitting in the car. In [20] it has been shown that the use of a predictive display can mitigate the impacts of lags by representing the latency based state, e. g. foreshadowing the time delay based on the car position. In [21] various types of predictive displays have been compared in a study, showing that their usage can effectively assist the operator with his task.

A different suitable approach is the use of a free corridor, where the operator has to decide which path is taken by the car if the connection is lost [22]. These approaches are based on the situational awareness of the teleoperator. This situational awareness can be better achieved, if the teleoperator is aware of the relevant environment [23], e. g. by having a suitable display of relevant data.

A user-centered design approach for developing an interface for Teleoperated Driving is shown in [24], allowing to be adjusted by the operator. User studies regarding Teleoperated Driving have been carried out by various research groups. In [25] Liu et al. conducted a user study with state-of-the-art LTE network performance and a small-scale vehicle. They claim that Teleoperated Driving over LTE does not work without supporting systems. Vozar and Tilbury [26] conducted a user study to explore the effects of latency. It is shown that the path-following score decreases with higher latency. A further user study, not specific to Teleoperated Driving was conducted by Nielsen et al. [23]. They introduced a combined 3D view and analyzed the results, showing that their approach improves the driving. Another user study was carried out in [27], where the stream quality was analyzed and showed an impact on the objective situation awareness. It was additionally shown that participants were able to identify important objects and maintain situational awareness in different driving situations on video streams with different qualities and display types.

Most of the previous work did not or only secondarily address the issue of the required bandwidth. In the research present in [28], the researchers were able to reduce the bandwidth-requirements to about 15 kbps, by transmitting a reduced LiDAR point-cloud, limiting the driving speed to about $5\,\mathrm{km/h}$ in a specific use case (road side work). In [19] the authors claim that for transmitting a field of 150° about 3 Mbps are required. Gnatzig et. al [16] present an approach where, based on heuristics, the compression parameters are updated with respect to the available bandwidth. For their *driv-*

ing relevant front-camera [16] they present two compression setups. The first with a resolution of 640x480, CRF 25 and H.264 bandwidth, and the second with 320x240, CRF 30 on H.264, which led to 1678 kbps and 222 kbps, respectively. Nevertheless, based on the findings in [29], this quality might not be feasible for real-world scenarios, i. e., the ability of only applying different compression parameters on a single video-stream is limited – especially if different driving situations are taken into account [29].

To address the drawbacks of previous works, an approach to reduce the required bandwidth by keeping all important environmental information is presented and supported by a user study.

III. Methodology

In order to lower the bandwidth requirements for Teleoperated Driving, this paper investigates different approaches which built on top of each other. The main idea consists of splitting a single video stream into two streams to separate important objects such as the driving lane and significant objects from the less important rest. Different filters and compression methods are applied to these streams. Finally, the two streams are merged and encoded prior to transmission.

At first, the most basic approach of separating the video-stream into two streams is presented. The basic camera stream is split into the driving lane in front of the vehicle, in the following called *mask*, and the *remainder*, i. e., everything else. For the experimental setup, the driving lanes for the different scenarios are annotated by hand to also include broader areas if turnings or lane changes happen. However, in real-world scenarios lane-detection systems such as the one presented in [30] would be used. In addition to the separation into two parts, a bilateral filter is applied to the *remainder* to maintain important edges, but remove unnecessary details on surfaces [31]. This approach allows – in combination with the H.265 compression – for a greater compression and lower bandwidth requirements.

Subsequently, this approach is enhanced by applying two different machine learning (ML) models (SSD MobileNet v2 320x320 and EfficientDet D7 1536x1536 from ModelZoo [32]) that perform object detection for objects that may become important for the current driving situation, e. g. pedestrians, other vehicles. In this case, the *mask*-part is enhanced by inserting important objects such as pedestrians, other vehicles, stop signs and traffic lights, etc. that are relevant for the selected scenarios as presented in Figure 6a. They will stay unchanged and allow for perceiving more details by keeping the bandwidth requirements low. The two ML models differ in their speed and accuracy and allow an estimate for real-world utilization under different initial conditions.

In order to advance the object-detection approach, a field of view, inspired by 360° videos [33], is defined, allowing the system to blur areas outside the field of vision stronger than the other parts of the stream. Blurring in the context of this paper means applying the bilateral filter to the raw image and not playing around with encoder settings, as the this fits better into the processing chain. This approach keeps the *mask*-part with lanes and – based on the approach – important objects, but reduces the bandwidth requirements of the *remainder* part.

All the above approaches have in common, that the important area in front[1] of the vehicle (driving lane) is never blurred and all details are kept. For the blurring, two different options are investigated. One approach (blur-full; BF) keeps the color in the *remainder*, while the other approach (gray blur-full; GBF) turns the *remainder* into gray and blurs afterwards. In general, the final videos were compressed with the individual parameters (resolution, tune/preset, crf and bitrate) that were identified as scenario-dependent driveable by [29]. Nevertheless, further specific encoding parameters, that can be used to fine-tune the bandwidth requirements by not altering the visual quality, are investigated.

In summary, the following sections present quality perceiving codec-parameters to achieve the lowest bandwidths. This is enhanced by discussing the lane-only approach, where only the lane is kept unblurred, while the rest is blurred. An advancement of this approach is adding important objects, which are identified by machine learning. Finally, a field of view is introduced in order to further reduce the bandwidth requirements. Resulting video clips are presented to participants in a user study, whose results were considered for an adaptive system.

A. Prerequisites

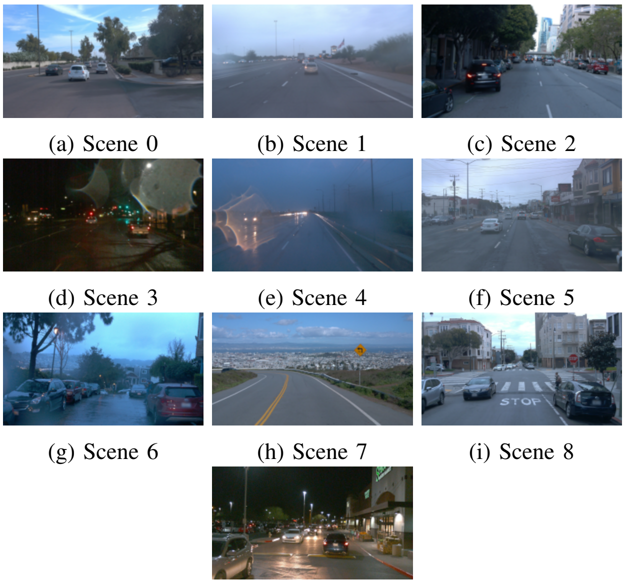
Allowing for a meaningful comparison of the obtained results, the video clips utilized for this paper are the ones that were used by Neumeier et al. in [29], consisting of a diverse set of traffic scenarios incorporating various environmental conditions. They were evaluated by a user study comparing different levels of quality based on codec adjustments. Additionally, the bandwidth bounds for a stream in which a scenario was considered as remotely controllable already exist for those scenarios. This allows to work with a baseline that needs to be undercut in order to make the new approach useful. The screenshots of the different scenarios can be seen in Figure 1.

The results of Neumeier et al. [29], addressing the visual quality of videos, indicate a broad range for the bandwidth requirements – based on different applicable compression parameters. The lowest number is 280 kbps for scene 0, while the upper bound is undefined for the two scenes 3 and 4, where none of the presented qualities were rated driveable (Table I). Their values will be assumed with optimistic 1000 kbps (based on the recommendations of YouTube [36] and Adobe [37] for sufficient streaming bandwidth) for this paper to not overestimate the effect of the applied approaches. However, real-world values might need to be somewhere around 3346 kbps (scene 3) and 1044 kbps (scene 4).

B. Dataset

The process of generating the video streams for the analysis in this paper consists of reading the images of the scenarios, generating new images based on the applied filters and writing

[1] In this work only one screen is considered, however other work ([6], [34]) addresses this topic.



(a) Scene 0 (b) Scene 1 (c) Scene 2 (d) Scene 3 (e) Scene 4 (f) Scene 5 (g) Scene 6 (h) Scene 7 (i) Scene 8 (j) Scene 9

Fig. 1: Scenarios that were used for the bandwidth optimization. (Source: [29] based on [35]), including scenarios 3 and 4 that are not considered for the user study.

Scene	Min Bitrate (kbps)	Scene	Min Bitrate (kbps)
0	643.81	5	831.92
1	280.00	6	698.29
2	739.58	7	570.82
3	Undefined	8	687.23
4	Undefined	9	299.20

TABLE I: Minimal required bandwidths (study compression) in kbps based on the results in [29], where only encoder settings were adjusted.

them back onto the disk lossless. Finally, these images are read by FFMpeg to generate videos with different parameters. For a meaningful comparison, the FFMpeg compression parameters which were identified as sufficient in [29] are applied for the final stream, consisting of *mask* and *remainder*. This ensures, that the compression does not work in a way that would manipulate the *mask*-part stronger as already being identified as lower bound.

The overall calculated data is about 313 GB, consisting of about 423.400 calculated images and 73.120 calculated video clips upon these images. The accumulated execution of generating all of those combinations took more than three weeks on an Ubuntu 20.04 system with an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz and 64 GB of RAM running on 10 parallel threads. OpenCV is used in version 4.2.0+dfsg-5, Tensorflow is in version 2.3.1. The applied ML models are *ssd_mobilenet_v2_320x320_coco17_tpu-8* and *efficientdet_d7_coco17_tpu-32*, both received from ModelZoo [32]. FFMpeg is in version (7:4.2.4-1ubuntu0.1).

C. General Discussion of Overall Results

In order to be able to compare the results of the following approaches, a basic introduction to the ideal parameters for the

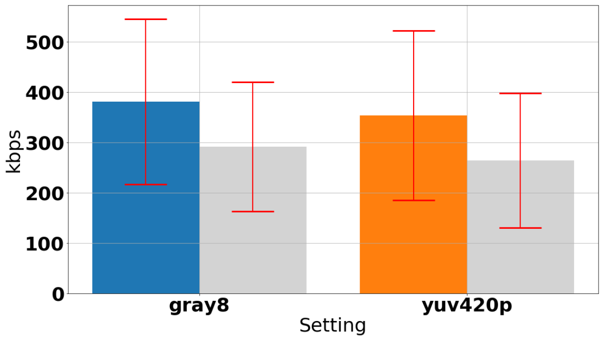


Fig. 2: Comparison of colorspace *gray8* and *yuv420p*. The gray bar indicates the median results for the *remainder* in GBF, while the colored area indicates the median bandwidth required for BF. The red error-bars indicate the standard deviation.

compression is required. These results cover codec parameters that can be adjusted to reduce the required bandwidth without affecting the perceived visual quality. The adjusted parameters are the motion estimation search method, the motion estimation search range and the colorspace comparison between 8 bit gray and colored streams [38]. The pre-defined scenario-dependent compression parameters (resolution, tune/preset, etc.) are not changed. H.265 is used for video compression in all cases.

Although some visual information might be lost, the first investigation was about whether transmitting a stream in the *gray8* colorspace could further reduce the overall required bandwidth in critical situations.

In contrast to expectations, the bandwidth increases by about 10% when utilizing *gray8* for compressing the stream, i. e., the values for the colored-blurring (BF) increase from a median of 353 kbps at *yuv420p* to a median of about 380 kbps for *gray8*. For the gray-blurring (GBF) the increment is about the same and needs to be investigated further in future work. Figure 2 shows the results for the *gray8* and *yuv420p* colorspaces. Based on these findings, the following analysis will only focus on video compression with the *yuv420p* colorspace.

Another parameter that keeps the visual quality untouched, but may influence the resulting bandwidth, is the motion estimation search method. In order to get an overview of the performance of the different search methods in the present scenarios, the following values are explored: *hex* (H.265 default), *umh*, *star*, *sea* and *full* (cf. Figure 3), covering all but the diamond (Dia) search method. The first ones are the fastest, while the last one is the slowest based on this order [38].

Hex consists of a similar approach as Dia, which starts "at the best predictor, checking the motion vectors at one pixel upwards, left, down, and to the right, picking the best, and repeating the process until it no longer finds any better motion vector." [39] Unlike Dia, *hex* "[...] uses a range-2 search of 6 surrounding points[...]." [39] *Umh* in H.265 "[...] is an adaption of the search method used by x264 [...]" [38] and "[...]searches a complex multi-hexagon pattern in order to avoid missing

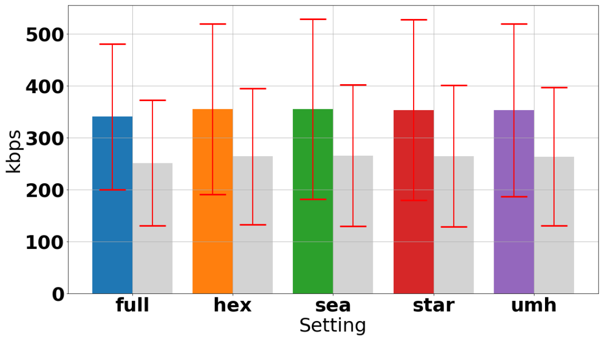


Fig. 3: Comparison of the different motion-estimation search methods on *yuv420p*. The gray bar indicates the median results for the *remainder* in GBF, while the colored area indicates the median bandwidth required for BF. The red error-bars indicate the standard deviation.

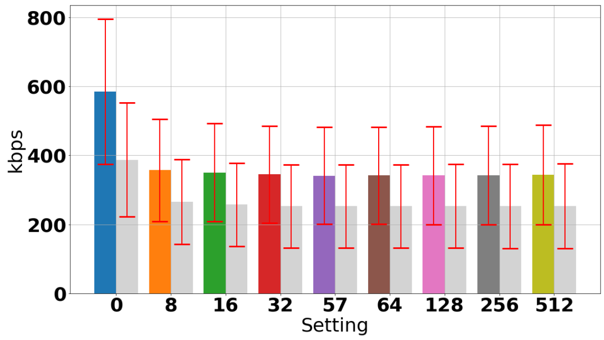


Fig. 4: Comparison of the different motion-estimation search ranges for *yuv420p* and *umh*. The gray bar indicates the median results for the *remainder* in GBF, while the colored area indicates the median bandwidth required for BF. The red error-bars indicate the standard deviation.

harder-to-find motion vectors." [39] "*Star* is a three-step search adapted from the HM encoder: a star-pattern search followed by an optional radix scan followed by an optional star-search refinement. *Full* is an exhaustive search; [...]. *SEA* is [...] a speed optimization of full search." [38]

The median bandwidth in BF ranges from 340 kbps for *full* to 354 kbps for *hex* and *sea* and as such has a variance of about 4% between the best and worst median results.

Although the *full* parameter result in the best compression ratio, the overall speed of the exhaustive search is too slow to be used in a system with strong latency requirements, e. g. about 7 fps in contrast to about 33 fps for *umh*. In real-world applications this conservative estimate on the achievable fps can change if using specialized hardware. Nevertheless, with a slightly greater bitrate than *full*, *umh* as the second best result at 352 kbps and acceptable performance of about 33 fps will be used for the rest of this paper.

The last parameter that is adjusted for the video compression covers the motion estimation search range. The values are changed between *0, 8, 16, 32, 57* (H.265 default), *64, 128, 256* and *512* to cover a broad range of meaningful values. Higher values are not tested as their execution is too slow, e. g. *1024* achieves about 10 fps in average while *256* reaches about 25 fps. The results for different search ranges on the setting *yuv420p* in combination with *umh* can be seen in Figure 4. The median values are 341 kbps for multiple ranges to 584 kbps for the range *0* in BF. As *57* is the default value of H.265 and results in the same bandwidth requirements as greater search ranges, which are slower, *57* will be considered as the search range utilized in the rest of this paper.

Although there is a combination of motion estimation search method and motion estimation search range that will lead to lower bandwidth requirements than the selected combination of *umh* and *57* by keeping tight time constraints for every single scenario, the rest of the paper considers this setup, as it leads to the best overall median results (all scenarios and all approaches are explained later). Future work will address this topic by developing an algorithm which selects the best combination of parameters depending on the current situation.

D. Manipulating the Stream to Reduce Bandwidth

The first very basic approach splits the single stream into two parts consisting of the *remainder* (Figure 6b) and *mask* (Figure 6a), where the red area indicates the area which is transmitted for the lane-only approach while the green area indicates additional embedded objects detected by ML techniques, which will be explained later. The idea behind this approach is that the most important driving-related objects are in the driving direction of the vehicle and this objects must stay above a certain visual quality – e. g. as the one identified by [29] – to be driveable by human operators, while less important areas of the video stream are note required to stay above such a level. After manipulating the two parts of the stream independently, both are combined again (Figure 6c), allowing the operator to perceive important objects in front of the vehicle. In order to not only compress the image in a simple way, i. e., pixelation, a more complex filter is applied, the *bilateral filter* of OpenCV (with the settings *diameter* = 25, *sigmaColor* = 125 and *sigmaSpace* = 250 [40]. On a NVIDIA RTX 2070 [41] with OpenCL [42], the whole process takes about 0.008 seconds (about 125 fps), while 0.00019 seconds are for masking and 0.001 seconds are for not optimized memory exchange from and to the GPU.). The basic idea behind this filter is that less important details are removed while the more important edges are preserved. To further reduce the bandwidth, this approach can be enhanced by removing the colors of the *remainder*, keeping it only as gray values. When textually describing the improvements in the following, the average improvements of all ten scenarios are presented, as this reflects the capabilities of the approaches the most, i. e., working under different environmental conditions. In order to simplify reading, absolute values are not presented in the following text, but are included in detail in Table VI at the Appendix. The results of this lane-only approach can be seen in Figure 5, indicated with the colors purple (BF) and maroon (GBF). The horizontal red lines represent baselines using traditionally compressed streams by the work of Neumeier et al. [29]. It can be seen that the results of the compression methods proposed by this work fall bellow these

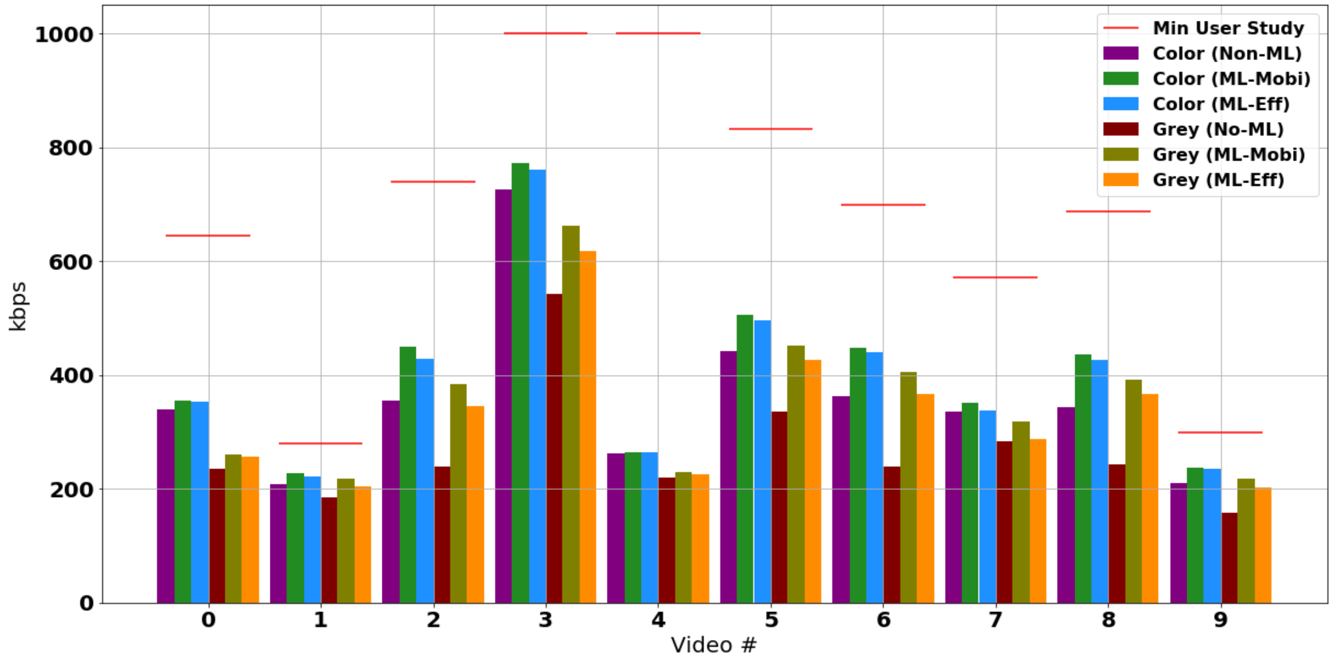


Fig. 5: Comparison of the lane-only and ML results for *yuv420p*, *umh* and a search range of *57*. The red bars indicate the bandwidth-requirements identified by [29] (Table I).

bandwidth baselines for each video. As such, it can be said, that the approach can help to reduce the bandwidth required for the stream. The average streams are 53% (BF) and 40% (GBF) of the original size.

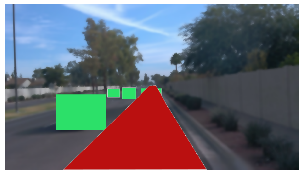
E. Applying Machine Learning

In addition to including the lane in front of the vehicle into the *mask*, other possibly important objects should remain visible for the remote operator. Objects and traffic participants like vehicles or pedestrians could also be relevant for safely guiding the vehicle remotely. As such, they should not be blurred but stay visible. The basic idea of this approach is shown in Figure 6, indicated by the green areas. Blurring only the *remainder* (Figure 6b) is also applied in this approach, i. e., the stream is combined before being transmitted (Figure 6c).

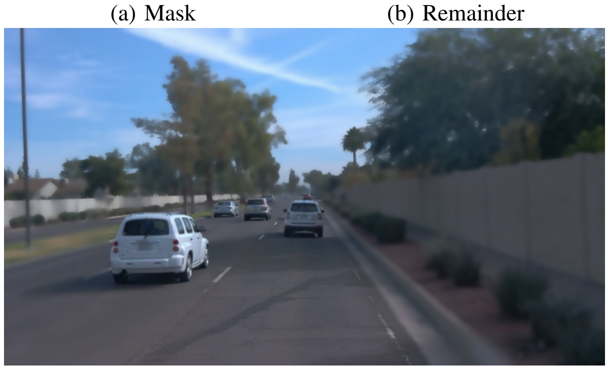
This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization.

In order to produce meaningful results, two different well known models are applied. They are chosen by their speed in FPS and their mean average precision (mAP; typically based on the intersection over union (IoU) across all classes) on the COCO dataset containing labeled and located objects in complex everyday scenes [43]. The slow EfficientDet D7 1536x1536 with a COCO mAP of 51.2 and a speed of about 3 fps (0.33 seconds per frame; without blurring, etc.) and the fast SSD MobileNet v2 320x320 with a COCO mAP of 20.2 and a speed of about 52 fps (0.019 seconds per frame; without blurring, etc.) following the results of [32]. EfficientDet achieved the highest COCO mAP of the list, while SSD MobileNet was the fastest but most inaccurate one. This setting allows to have an efficient comparison of slow but accurate and fast but inaccurate models at their extremes.

(a) Mask



(b) Remainder



(c) Combined

Fig. 6: Example of the approach to split into mask (a) and remainder (b). The area indicated by red is the one of the mask with the lane-only approach, while green together with red indicates the ML approach area. (c) shows how the stream will be be transmitted finally.

To get an overview of how accurate or inaccurate the pretrained detection models are, Figure 7 shows a comparison between both approaches. The blue and green marked areas

Fig. 7: Difference between the to ML approaches. Blue indicates areas detected by MobilNet only, green indicates areas detected by EfficientDet only.

Scenario	# MobileNet	# EfficientDet	% Diff
0	7.49	4.13	55.10
1	7.13	5.98	83.92
2	22.39	15.33	68.47
3	6.26	2.95	47.18
4	0.10	0.56	555.00
5	17.90	12.49	69.76
6	9.49	8.34	87.87
7	1.12	0.29	25.68
8	10.71	9.80	91.56
9	18.09	8.93	49.36

TABLE II: Comparison of the average count of detected objects between MobileNet and EfficientDet, calculated by counting detected objects per frame and dividing that number by the count of frames.

indicate the objects exclusively detected by each ML approach. It can be seen that the identified objects and their specific areas differ substantially, which will lead to a difference in the display of important objects.

In addition, Table II shows the count of recognized objects per method averaged over the whole scene, which helps to determine the overall detection capabilities. The last column represents the difference in percent of detected objects from both methods. For all scenarios and both models, the detection threshold was set to 0.45, i. e., the model is confident to 45% that an object was detected and classified correctly. Although this seems to be a low value, the system is safer if transmitting more uncertain objects than missing one important one.

It can be seen that the average difference in the count of detected objects ranges between 25.68% and 91.56% if the very high value of 555% is neglected. This high value can be explained by the fact, that scenario 4 has very bad light and weather conditions and thus the detection is very inaccurate, which means that the operator needs to react accordingly.

*1) **SSD MobileNet v2 320x320**:* The *SSD MobileNet v2 320x320* model was the fastest but also the least accurate in the ModelZoo [32]. The results of this model within the paper application can be seen in Figure 5, indicated by green (BF) and olive (GBF). The average results are 63% (BF) and 56% (GBF) of the original bandwidth requirements. In comparison to the approach without the usage of ML, the introduction of further objects lowers the overall improvement. Compared to the approach where only the lane is ignored from blurring, the average savings are 46 kbps (BF) and 86 kbps (GBF) lower than without machine learning.

Fig. 8: Field of view as used in the proposed approach.

*2) **EfficientDet D7 1536x1536**:* With *EfficientDet D7 1536x1536* the most accurate model in the ModelZoo [32] was chosen. Results of this model can be seen in Figure 5, indicated by blue (BF) and orange (GBF). The average required bandwidth for BF and GBF compared to the original required bandwidth are 62% and 52%, respectively. In contrast to the scenario where no ML was applied, the average bandwidth improvement is lower to the extend of 38 kbps (BF) and 62 kbps (GBF), but better than the ones using the SSD MobileNet model. EfficientDet in average requires 9 kbps (BF) or 24 kbps (GBF) less than the SSD MobileNet approach.

F. Applying Field of View

Based on these straight forward improvements, an enhanced approach is applied to further reduce the required bandwidth. The approach addressing the field of view (fov) is based on the assumption, that primarily the center of an image is perceived sharply by humans, while everything in the outer area can not be focused simultaneously. Solely the important center of the image is focused and hence sharp, while everything out of this area is blurred with the bilateral filter of OpenCV (*diameter* = 200, *sigmaColor* = 225 and *sigmaSpace* = 250 [40], which leads to about 0.03 FPS (about 30 seconds per frame) using OpenCL [42] on a NVIDIA RTX 2070 [41].). An example of this can be seen in Figure 8. This approach, applied for 360° videos [33] via encoder settings, is based on the assumption, that important objects should be displayed as sharp as possible, allowing the remote operator to perceive them optimally. The application of this approach is threefold: In the first stage, the area out of the field of view is blurred with a very strong blurring. The area within is blurred with the same values as applied in the approaches above. The used driving lane itself is never blurred and stays as sharp as possible. Furthermore, this approach will also be enhanced by the two already introduced ML approaches and by that exclude important objects from the blurring process.

Results as can be seen in Figure 9 indicate that this approach with lane only can further lower the bandwidth requirements for a stable and safe remote connection. On average, it reduces the required bandwidth to about 44% (BF) and 34% (GBF)

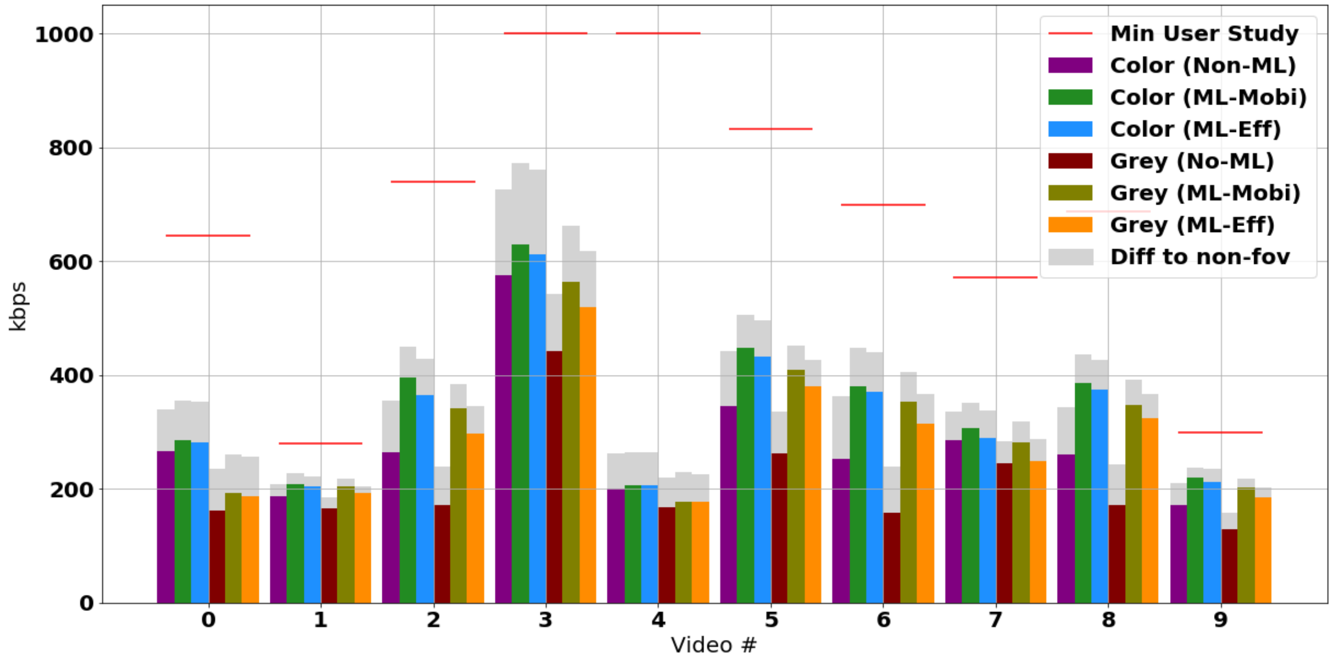


Fig. 9: Comparison of the field of view lane-only and ML results for *yuv420p*, *umh* and a search range of *57*. The red bars indicate the bandwidth requirements identified by [29] (Table I). The gray bars show the difference between the non-fov (gray) and the fov (color) approach.

of the original bandwidth. The improvement in contrast to the non-fov lane-only approach lies at about 77 kbps (BF) and 60 kbps (GBF).

For the field of view, both the SSD MobileNet and the EfficientDet are applied to identify and incorporate important objects to the streams.

*1) **SSD MobileNet v2 320x320**:* The results of SSD MobileNet (cf. Figure 9) show an average improvement of 55% (BF) and 49% (GBF) from the original required bandwidth. In comparison to the lane-only field of view approach – without important objects – the average is 65 kbps and 100 kbps greater for BF and GBF, respectively. By comparing the results with the same model but not using the field of view approach, there is an average improvement of 58 kbps (BF) and 47 kbps (GBF).

*2) **EfficientDet D7 1536x1536**:* Additionally, the results of the EfficientDet model were also utilized for the field of view approach. The overall average required bandwidths are 53% for BF and 45% for GBF (cf. Figure 9). In accordance with the field of view approach and the SSD MobileNet model, the average bandwidth is also greater than without machine learning. Nevertheless, the model's average required bandwidth is about 12 kbps (BF) and 25 kbps (GBF) below the requirements of the SSD MobileNet model. In contrast to the same EfficientDet model but without the field of view approach, the average improvements are 61 kbps (BF) and 48 kbps (GBF).

IV. User Study

In order to be able to utilize the presented approaches in real-world applications, not only the bandwidth reduction is important, but also the real-world applicability based on the perceived quality and the related trust in a specific setting. This can be, for example, evaluated by human ratings on driveability and perceived video quality for the distinct settings. Therefore, a user study was conducted where participants had to rate the driveability and the perceived video quality.

A between-subjects user study using the online-service of SoSci Survey [44] is conducted. The study is designed to be finished in about 7 - 10 minutes. Participants have to rate the perceived video quality (for Mean Opinion Score (MOS), 5-Point Likert Scale) and the driveability (4-Point Likert scale) of various video clips. The MOS is chosen as it is a widely known and well understood practice to measure the perceived quality of media [45], i. e., the study design thus fits the ideal sequence length of 8 s – 10 s for the stimuli as proposed in [46]. Every participant is shown $n = 20$ different randomly chosen video clips S_n out of the total $N = 192$ available ones S_N, thus $S_n \subseteq S_N$. The n video sequences consist of all available combinations with the previously mentioned optimizations, i. e., a combination is a tuple $(scenario, fov, color, ml)$ consisting of all potential combinations per scenario, ignoring scenario 3 and 4 and applying the previously explained *umh*, *57* and *yuv420p* encoder settings. However, if referred to a specific scenario, the tuple is consisting of $(fov, color, ml)$. The (random) selection process is designed to achieve a uniform distribution of ratings per video and is based on random sampling without replacement. Two types of compression are applied: *study compression* based on the compression settings leading to minimal bandwidth requirements as identified by [29] (Table I) and *basic compression* with the parameters resolution, present and tune set to 1600x900, ultrafast and fastdecode, respectively.

The scenarios 3 and 4 are not part of the user study, as the results in [29] already indicated that even the *basic compression* did not lead to a rating one would regard driveable, i. e. even the best quality presented to the participants was rated not suitable for remote driving. Such critical situations can be avoided be planning the drive accordingly, i.e. enhancing the area whitelisting approach shown in [12] with weather and light conditions.

The online survey itself starts with a page introducing Teleoperated Driving, so that all participants know the basics of such a system and have the same level of understanding. This introduction was then followed by the instructions on how to conduct the user study stating that only participants with a valid driver's license are allowed to participate, avoiding total color blind participants. Afterwards, the selected 20 video clips are presented sequentially to the participants. Based on the provided tasks *"Please rate the perceived quality of the video-clip seen just now."* and *"Would you rate the perceived quality as sufficient for Teleoperated Driving?"* the participants have to rate the MOS and the driveability. The options to answer regarding MOS are *Excellent*, *Good*, *Fair*, *Poor* and *Bad* [47], while for the driveability they are *No*, *Rather No*, *Rather Yes* and *Yes*, avoiding the possibility to rate *Uncertain*.

A. Dataset

The user study was online for about one month in 2021 and participants were gathered through distributing E-Mails with an invitation to participate at the user study and the online-tool Surveycircle [48]. All links were identical impersonal links to maintain the anonymity of participants. In order to be General Data Protection Regulation (GDPR) [49] compliant, no personal information about the participants, e. g. age, gender, was collected, as in [29] it turned out that there was no difference between gamers, non-gamers, gender, etc. In total about 320 potential participants opened the study link and clicked at least once on the *NEXT* button. Yet, only 268 participants finished the study, i. e., they rated all $n = 20$ presented video clips. 238 valid participants remain after filtering based on completion time. Participants with a study completion time below $250\,\mathrm{s}$ are removed as this duration would mean that they were voting without taking their time to properly watch and rate the videos. The duration for an attentive evaluation is regarded to be $t > 250\,\mathrm{s}$. As the participants were presented 20 videos with about a length of $10\,\mathrm{s}$ each, the remaining time for reading the introduction and rating the video would be $50\,\mathrm{s}$, which is deemed for not being sufficient for a thoughtful rating. A further reduction on the number of participants happens by removing users that conducted the study with smartphones, as these devices will distort the results due to their small screen (participants were informed to not use smartphones for conducting the study). This leaves a total of 226 valid and usable ratings of the participants.

The number of ratings per video vary between 16 and 30, which means that the video clip with least ratings is still above 10 ratings and thus can be used for the analysis. The median time for finishing the study was $475\,\mathrm{s}$, with a range of $252\,\mathrm{s}$ to $1486\,\mathrm{s}$.

Scenario	MOS	Scenario	MOS
0	2.62	6	2.17
1	2.23	7	2.49
2	2.56	8	2.51
5	2.21	9	2.13

TABLE III: Scenario-based average MOS for all video clips with *study compression*.

B. Results

As a first general result, the overall driveability rating on all $N = 192$ video clips is *Rather No*, while the overall MOS is 2.5 and thus between *Poor* and *Fair* indicating a high Spearman correlation [50] of about 0.95 between both (average), which will be important in order to be able to use the MOS for providing sorted suggestions in the later explained proposal system. More specifically, 57 videos ($\sim$30%) out of the 192 were ranked as driveable, which means that the median ratings are at least *Rather Yes*, for the applied parameters as described in Section III. Performing the Kruskal–Wallis H test [51] with $\alpha = 0.05$, indicates that there is a significant difference between the individual scenarios, the fov settings, the ml settings, the color settings and the compression settings. In general 35 combinations ($\sim$61%) were ranked as driveable for the *basic compression*, while 22 combinations ($\sim$39%) were ranked sufficiently for the *study compression* settings.

However, as driveable rated video clips of the *basic compression* are about 693 kbps above the results of [29], as shown in Table I and Figure 6c, they were intended only as baseline for the case that a scenario has no driveable rated combination of settings for the *study compression*. Thus, the important results are the values of the *study compression*: Driveable rated video clips are in average about 247 kbps below the results of the user study in [29] (Table I) and at least one driveable combination exists for each scenario. For further investigation only these *study compression* video clips are considered. The overall median trust of the *study compression* video clips is *Rather No*, like for all video clips, while the MOS has decreased slightly to 2.37 compared to the 2.5 considering all video clips. The per scenario median driveability is always *Rather No* and the average MOS per scenario can be seen in Table III.

Although every scenario has at least one combination $(fov, ml, color)$ that is rated driveable, it turns out that there is not the one combination that fits all scenarios. In Table IV the parameter combination with the highest MOS being driveable for every scenario is shown, if multiple combinations were rated driveable. It can additionally be seen that scenarios have a different number of combinations that are rated driveable, e. g. 5 combinations for scenario 2, 1 combination for scenario 5 and so on. It is noteworthy that for all scenarios except scenario 5 at least one combination per scenario was with nofov. One thing that all driveable rated video clips have in common is, that at least one combination per scenario is with color. Considering the other not listed but driveable rated parameter combinations, it turns out that this are different combinations of fov, ml and color. Overall, the 22 driveable rated video clips for the *study compression* have the settings

Scenario	FOV	ML	Color	#	Improvement (kbps)
0	nofov	mleff	col	4	317.98
1	nofov	mleff	col	1	58.31
2	nofov	mlmobi	col	5	345.1
5	fov	mlmobi	col	1	384.26
6	nofov	mlmobi	col	2	291.0
7	nofov	mleff	col	4	242.8
8	nofov	mleff	col	3	277.4
9	nofov	mlmobi	col	2	62.82

TABLE IV: Drivable combinations per scenario. If there is more than one driveable combination per scenario, the one rated with the highest MOS is shown by example. The bitrate improvement in kbps reflects the average improvement across all driveable combinations for the specific scenario. The # indicates the number of driveable combinations for that scenario.

nofov (20) – fov (2), mleff (12) – mlmobi (8) – noml (2) and color (16) – gray (6). Although only 10 different scenarios are considered, it can be seen that selecting the ideal combinations of parameters by hand can become hard already.

V. Adaptive System Design

In order to support the remote operator during the process of choosing the most suitable combination out of those potentially different driveable combinations, a strawman system is presented. Before introducing this system in detail, an analysis to determine specific preferences of the user study participants regarding the combination of the different combinations $(fov, ml, color)$ of all rated clips is carried out. Preferences in this case means that the participant's ratings with this specific combination were always above the comparable average of the participant's rating, i. e., the specific combination *(color, mleff, nofov)* was rated above the average and the specific *color*, *mleff* and *nofov* were also rated as individual parameter above average.

It turns out that about 56 out of 226 ($\sim$25%) participants have a preference on a specific combination, while 14 of them even have two preferred combinations. Every combination of two preferences has only one difference: the usage of mleff or mlmobi. All other parts of the combinations are the same if a participant has two preferences. This needs to be considered for the system design, as individual remote operators may feel more safe driving a specific combination.

In general, the adaptive system selects the ideal combination $(fov, ml, color)$ and codec parameters to filter the video stream and, hence, reduce bandwidth requirements by taking into account the prevailing environmental conditions. The main idea stems from the observation, that different environmental conditions in the video clips led to different driveable rated combinations. The videos differed in the infrastructural aspects (*rural, urban, suburban*), weather conditions (*sunny, rainy, foggy*) and light conditions (*day, night, sunrise*). Based on the computation of the available bitrate and the results of user studies, this helps to define a lookup-table[2] suggesting the ideal combination $(fov, ml, color)$ for the given environmental conditions and the accordingly used codec parameters for achieving the combination. Currently per scenario only one set of *study compression* parameters exist per scenario (see Table I for bitrates) and thus the focus is mainly on the new parts of the approach as explained previously. The idea is not that the system automatically selects a combination $(fov, ml, color)$, but the remote operator can chose from a presented number of combinations, e.g. five combinations in the following.

[2]Such a lookup table could be continuously refreshed and updated based on the teleoperator's feedback and driving performance, e. g. by applying ML.

The central part of the proposed system design is the lookup-table, which consists of all combinations of environmental conditions, approach parameters (e. g. fov, ml, color), driveability rating, MOS and the target bitrate under the given conditions. An example of such a table can be seen in Table V and usually needs to be build only once and then can be used whenever it is required to check for a specific configuration. A second table could be used to map the scenarios to specific codec settings. For more complex setups, e. g., different codec parameters for the same scenario, this can be combined into one table, but this approach is not explained further.

The content of the table can be built as done within this paper by determining all different types of combinations and presenting them to a sufficient number of participants, which then rank for driveability and perceived quality. Additionally, it makes sense to include future remote operators to rate the driveability and perceived quality in order to check whether they have individual preferences on specific combinations.

Based on this knowledge and the determination of the available bandwidth, the algorithm presented in Figure 10 can be used to predict the ideal combination $(fov, ml, color)$ and the codec parameters for the current environmental conditions. The algorithm requires the input of the available bandwidth, the current operator and environmental conditions: Area, Weather and Light. At first it checks whether the available bandwidth is above the *study compression* values (as in Table I) and if so, it does not need any specific further combination. The algorithm will return basic codec settings only. If the available bitrate is below the *study compression* values, the advanced approach is pursued, but the codec parameters remain the scenario-specific ones. Therefore, the approach selects combinations that match the given environmental conditions, are rated driveable and require less than the available bandwidth for transferring the video stream. If multiple combinations are found, they are sorted based on the remote operator's preferences firstly and on the rated MOS secondly. In order to facilitate the selection process, the number of printed results is limited to the five best feasible options. To also be able to deal with situations in which less than five combinations are rated driveable, the remaining entries ($5 - k$ already selected combinations) will be filled using the entries with the largest MOS, sorted by the operator's preferences. However, this will be marked with a hint, that the driving speed needs to be reduced. If none or less than 5 results exist, combinations with greater bitrate requirements will be presented with a hint, starting ascending with the lowest available bitrate. In general, returned combinations can be in either one of the groups *driveable*, *potentially driveable with speed adjustment* or *above available bandwidth* if below

Area	Weather	Light	FoV	ML	Color	Drive	MOS	Bitrate (kbps)
suburban	sunny	day	fov	eff	col	2.0	2.76	281
suburban	sunny	day	fov	eff	gre	2.0	2.0	186
suburban	sunny	day	fov	mobi	col	2.0	2.87	285
				⋮				
urban	rainy	day	nofov	mobi	gre	2.0	2.0	406
urban	rainy	day	nofov	noml	col	2.0	2.29	363
urban	rainy	day	nofov	noml	gre	1.0	1.38	240

TABLE V: Example of a simple lookup table consisting of the input parameters in gray and the potential resulting combinations in green. Codec parameters are not considered for demonstration purposes.

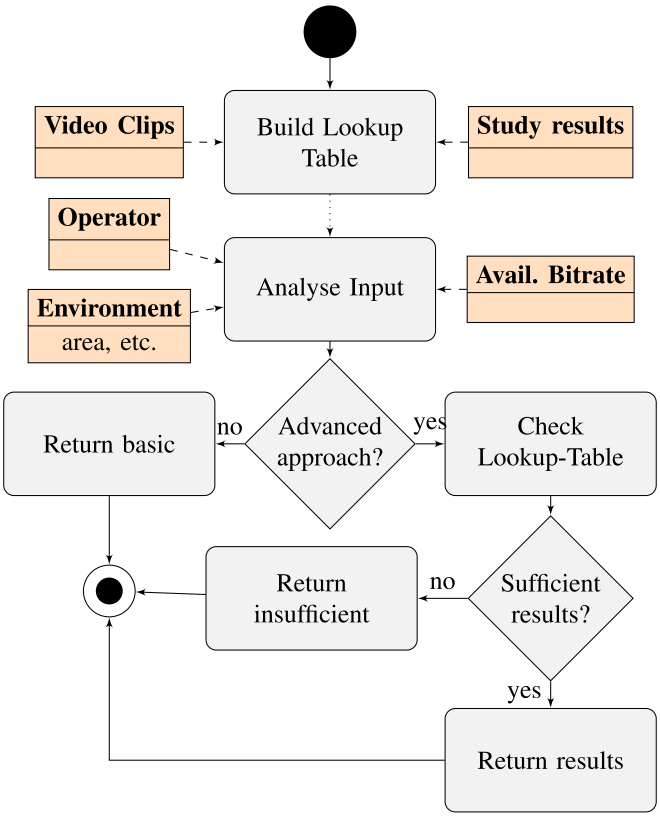


Fig. 10: Flow diagram of the algorithm from building the look-up table and generating the proposed combinations based on the given environmental conditions and the available bitrate. The process usually starts at the node *Analyse Input* if a lookup-table was defined priorly.

the *study compression*.

Finally, an application example of the proposed system is sketched. It introduces a typical use case by consulting the available data of the user study, presuming that the operator has no personal preferences. Any possible personal preference, however, would only influence the sorting of the results if multiple exist but not their grouping. As a basic scenario the environment consisting of *subrban*, *sunny* and *day*, e. g. scenario 8, is used. For presentation purposes, the available bitrate is altered between 300 kbps, 600 kbps and 1200 kbps. With 1200 kbps as available bitrate, the system proposes the *study compression* as operating settings, i. e., 1200 kbps are above the required 687.23 kbps of the codec-only approach (cf. Table I). Compression values could be obtained from the specific settings, e. g. resolution: 1280x720, preset: ultrafast, tune: fastdecode, crf: 30 in this case as indicated in [29]. For setting the value to 600 kbps, the system proposes three drivable settings and lists them, e. g. $(nofov, mleff, col)$, $(nofov, mlmobi, col)$, $(nofov, mleff, gre)$. It further lists two additional settings that might be driveable, but require a velocity reduction, e. g. $(fov, mleff, col)$, $(nofov, noml, col)$. If specifying the available bitrate with 300 kbps, the system presents three results that might be driveable with reduced velocity, e. g. $(fov, noml, col)$, $(nofov, noml, gre)$, $(fov, noml, gre)$ and two further combinations of parameters with bitrates above the specified available bitrate, e. g. $(fov, mleff, gre)$, $(nofov, noml, col)$. The last two examples with 600 kbps and 300 kbps would have the same codec settings as the one with 1200 kbps, as only one *study compression* setting exists. All those values are supported with additional information such as the rated driveability, the MOS and the required bitrate for the specific approach. For presentation purposes these values are removed.

Based on those results, a remote operator can chose the most suitable approach and select an individually preferred combination of parameters, which will then be combined with the already known codec parameters. For realizing such a system in the real-world, it is important that with changing networking parameters, the change between different combinations is smoothly, i. e., the operator notices the transition only marginally and not from one second to another.

VI. Limitations

Although a variety of different combinations and approaches were presented and the user study had more than 200 participants, this work has its limitations. The first one is the limited number of only 10 video clips. Even if being selected to cover as many real-world scenarios as possible, the coverage is far from being exhaustive. Another major limitation is that only a narrow number of combinations was tested. With different blurring parameters other and maybe even greater improvements could be achieved. However, for this paper multiple different blurring parameters were applied and more can be gathered by further testing and expanding the system. The selection of the parameters was carried out based on visual selection. The selected combinations still allow the sensing of the blurred environment in the majority of the cases. Although being limited to one front camera, the applied approach could easily be extended to use multiple cameras, e. g. as with the combination of multiple streams into one as shown in [34].

The user study has its limitations in the restricted number of video clips, their short length and the limited number of participants. Nevertheless, the results can be used to support the claim of the paper, that the proposed approaches can work, as for every scenario at least on combination of parameters was rated driveable. With a large number of ratings per video clip, the drawback of different displays, on which participants watched and rated the video clips, could also be compensated, i. e., smartphones were already filtered beforehand. Equally, the proposed system design is limited by the number of video clips and user ratings, as more and different environmental conditions would be required to build a system that can be directly used generic. Yet, for the feasibility demonstration this in combination with the short length of the video clips is not a big deal as it shows that such a system can work. However, it can only be used for real-world applications when including more and longer video clips and a greater number of participants.

Selected ML models did additionally not track all available street signs, but only the most important ones for the specific scenarios such as stop-signs and traffic lights. However, this should not have a large impact on the study, as scenarios were selected properly to be used with the tracked objects. In addition, ML models are in general about to not detect objects, to misclassify them or to be tricked into something [52] and thus this may not be as reliable as one would like them to be. Nevertheless, the proposed approach always keeps the most important part sharp: the driving lane. Thus, this mainly impacts the available reaction time of the remote operator, which anyhow should be increased by speed reduction [53], or in future may be supported by additional sensors and improved ML models.

Finally, the application of filters and image preprocessing always adds cost in form of latency on the system, which is suboptimal for Teleoperated Driving. With the usage of specialized hardware such as modern autonomous driving boards like NVIDIA Drive AGX [54], which are powerful and capable of executing object detection/tracking in real time (based on the model), the latency impact can be reduced. Specialized and optimized algorithms that, e.g. are directly optimized for the target hardware can also help to further speed-up the execution, e.g. as shown for a CUDA-based bilateral filter which improved the performance about 600 times [55]. Further work such as [56], also stated that the major part of latency in traditional setups (only compressed stream) is mainly caused by network and monitor latency, less by the camera and processing. In addition, further approaches such as a slight speed-adjustment based on the system's latency [53] can be applied to allow for a safe drive, even if the latency is increased by the approach. However, there is an unavoidable trade-off between latency and bandwidth savings, but clever approaches help to lower the overall impacts on the system.

VII. Conclusion and Future Work

This paper presents a sophisticated approach to reduce the bandwidth requirements of a video stream in order to enable an operator to safely control a vehicle through Teleoperated Driving. The approach splits the original stream into two separate parts, consisting of *mask* and *remainder*. The *mask* contains all important objects to maneuver the vehicle safely. The *remainder* contains everything else. Based on that fact, it is possible to apply filters on the *remainder* to forego image details and instead gain a reduced video size which requires a lower streaming bitrate. In this paper the bilateral filter is applied that keeps edges but blur the image. Before streaming, both parts are put together and the typical encoder-based compression is applied. The goal of this paper is not to present a sophisticated real-world system that already chooses the best technique with respect to any given driving situation, or to provide an integration into congestion control but to present a reasonable approach, validate the results with a user study and present a system design that can be used for real-world applications.

With regard to the contributions, the results of the paper are the following: The results of **contribution 1** show an average bandwidth reduction of up to 467 kbps, which is about 34% of the original required bandwidth. The results of the user study – **contribution 2** – show, that for every tested scenario at least one combination $(fov, ml, color)$ was rated driveable, while the average bandwidth improvement across all driveable rated video clips is about 247 kbps. Based on the fact that different combinations were rated driveable for different scenarios, **contribution 3** proposes a system design that can be used to determine the ideal combination within distinct situations.

Overall it can be stated, that the proposed approaches can help to reduce the required bandwidth and as such help to enable Teleoperated Driving in greater geographical areas.

The first step in future work consists of the idea, that recognized objects might not be embedded into the stream, but are transmitted as objects in a separate stream. The advantage of this approach is that objects might not be transmitted every frame as they can be adjusted at the operators side based on factors like speed or distance. Although objects are transmitted separately, this of course cannot be applied for the lane in front of the vehicle, i. e., the lane mandatory needs to be embedded into the stream. First results indicate that the maximal available bandwidth per object is at a median of 12 kbps (MobileNet, gray), while the lowest available bandwidth is at a median of 7 kbps (MobileNet, color). This approach also allows for using additional sources such as Car2X-based data, e.g. for exchanging information of positions and velocities of other vehicles even beyond line of sight. Yet, this needs to be investigated further and validated via user study in order to check if the presentation of important objects as static parts in a stream can work as expected. However, the work of [57] embedded 3D objects identified by a LiDAR in their 360 degree stream and showed that this can support the driving task, which indicates a promising direction.

The approach of separating a stream into two parts could be taken further in future approaches, i. e., by transmitting the two parts as two independent streams. The *remainder* stream can then be manipulated differently and might not required the same framerate or have the same importance as the *mask* with the important objects. This can then be improved by utilizing additional sensor data, e.g. LiDAR-based data.

It is also important to address the trade-off between stability and agility. Future work will also address the question on how fast it is possible to switch between the normal operation and the proposed approaches of this paper. This will be combined with an algorithm that chooses the best technique and also considers for congestion control.

Finally, an approach where no stream at all is transmitted is considered. In this approach all important objects within the video stream would be tracked by a model (e. g. by applying ML) and transmitted as objects. This can help to lower the required bandwidth and limit the effects of latency as objects can be drawn dynamically in their real-world non-delayed position on the operator's side.

APPENDIX

Table VI shows the absolute bandwidth requirements of the individual combinations per scenario in kbps. Figure 11 gives an overview of all applied approaches to allow for a comparison based on *basic compression* videos presented to the participants.

Scenario	Blurring	Machine Learning (kbps)			Field of View (kbps)		
		None	ML-Mobi	ML-Eff	None	ML-Mobi	ML-Eff
0	BF	339.18	354.29	352.61	265.55	285.06	281.06
	GBF	235.33	260.61	257.22	162.74	192.26	186.44
1	BF	208.14	227.37	221.69	187.36	207.36	204.17
	GBF	185.73	217.46	205.26	166.10	205.02	193.47
2	BF	354.09	450.27	427.85	264.30	395.09	365.54
	GBF	239.61	384.03	344.72	170.90	340.67	297.54
3	BF	726.84	771.34	759.76	575.70	628.46	612.34
	GBF	542.25	661.58	617.43	441.33	563.53	518.70
4	BF	262.11	264.30	264.01	201.25	206.58	205.50
	GBF	219.41	229.13	226.38	168.59	178.16	177.27
5	BF	441.53	505.07	495.52	346.13	447.66	432.63
	GBF	334.86	452.02	426.99	262.84	408.67	379.22
6	BF	362.55	447.47	440.40	253.05	380.18	370.05
	GBF	240.01	406.04	367.10	158.49	353.25	314.83
7	BF	335.32	351.80	337.24	285.77	307.65	288.68
	GBF	284.20	317.96	287.71	244.60	280.97	248.03
8	BF	342.98	436.76	425.87	259.79	385.47	375.18
	GBF	243.44	391.11	366.87	171.63	347.38	323.67
9	BF	210.32	237.76	235.00	171.30	219.21	211.28
	GBF	157.55	218.25	203.08	129.18	201.74	185.86

TABLE VI: Absolute values of the specific approaches in kbps. The first row indicates the scenario, while the second one indicates the type of Blurring, i. e. either color (BF) or gray (GBF). The other rows show the distinction between Machine Learning (ML) and the Field of View (FOV) approach, either with one of the ML models or no ML applied.

(a) NOFOV NOML COL

(b) NOFOV NOML GRE

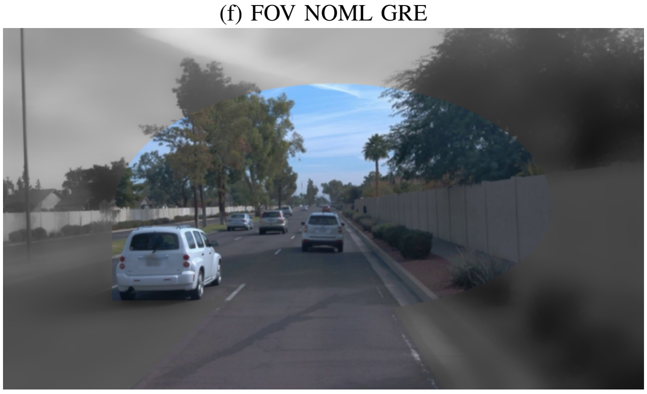
(c) NOFOV ML COL

(d) NOFOV ML GRE

(e) FOV NOML COL

(f) FOV NOML GRE

(g) FOV ML COL



(h) FOV ML GRE

Fig. 11: Comparison of all applied approaches in the order of their introduction within the paper. Images (a) – (d) show the non FOV approach, while images (e) – (h) show the approach with applied FOV. Images (a),(b),(e),(f) show the approach without the application of ML, while images (c),(d),(g),(h) show the utilization of the ML-Eff model.

REFERENCES

[1] T. Litman, "Autonomous Vehicle Implementation Predictions - Implications for Transport Planning," Victoria Transport Policy Institute Victoria, Canada, Tech. Rep., Mar. 2019.

[2] On-Road Automated Driving (ORAD) committee, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE International, Tech. Rep., Jun. 2018.

[3] A.-K. Frison, P. Wintersberger, A. Riener, C. Schartmüller, L. N. Boyle, E. Miller, and K. Weigl, "In ux we trust: Investigation of aesthetics and usability of driver-vehicle interfaces and their impact on the perception of automated driving," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: ACM, 2019, pp. 144:1–144:13. [Online]. Available: http://doi.acm.org/10.1145/3290605.3300374

[4] T. W. Victor, E. Tivesten, P. Gustavsson, J. Johansson, F. Sangberg, and M. Ljung Aust, "Automation expectation mismatch: Incorrect prediction despite eyes on threat and hands on wheel," *Human factors*, vol. 60, no. 8, pp. 1095–1116, 2018.

[5] L. Kang, W. Zhao, B. Qi, and S. Banerjee, "Augmenting self-driving with remote control: Challenges and directions," in *Workshop on Mobile Computing Systems & Applications*, 2018. [Online]. Available: http://doi.acm.org/10.1145/3177102.3177104

[6] M. Harris, "CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads," https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads, Jan. 2018, accessed on 28.11.2018.

[7] 5GAA Automotive Association, "Tele-Operated Driving (ToD): System Requirements Analysis and Architecture," 5GAA, Tech. Rep., Sep. 2021.

[8] A. Davies, "Nissan's Path to Self-Driving Cars? Humans in Call Centers," https://www.wired.com/2017/01/nissans-self-driving-teleoperation/, May 2017, accessed on 21.10.2018.

[9] Ericsson, "Remote operation of vehicles with 5G," https://www.ericsson.com/4add9b/assets/local/mobility-report/documents/2017/emr-november-2017-remote-operation-of-vehicles-with-5g.pdf, Jun. 2017, accessed on 10.03.2020. [Online]. Available: https://www.ericsson.com/4add9b/assets/local/mobility-report/documents/2017/emr-november-2017-remote-operation-of-vehicles-with-5g.pdf

[10] A. Davies, "The War to Remotely Control Self-Driving Cars Heats Up," https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars/, Mar. 2019, accessed on 04.04.2019.

[11] S. Neumeier, N. Gay, C. Dannheim, and C. Facchi, "On the Way to Autonomous Vehicles - Teleoperated Driving," in *AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium*. Dortmund, Germany: VDE, 2018, pp. 1–6.

[12] S. Neumeier, E. Walelgne, V. Bajpai, J. Ott, and C. Facchi, "Measuring the Feasibility of Teleoperated Driving in Mobile Networks," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, June 2019.

[13] K. Schilling, H. Roth, and R. Lieb, "Teleoperations of rovers. From Mars to education," in *Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on*, vol. 1, Jul 1997, pp. SS257–SS262 vol.1.

[14] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.

[15] A. F. Winfield, "Future Directions in Tele-operated Robotics," in *Telerobotic Applications*, 2000.

[16] S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp, "A System Design for Teleoperated Road Vehicles." in *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, Jul. 2013, pp. 231–238.

[17] J. Wannstrom, "Lte-advanced," *Third Generation Partnership Project (3GPP)*, 2012.

[18] Sichere Intelligente Mobilitaet Testfeld Deutschland, "Deliverable D5.5 – Teil A TP5-Abschlussbericht – Teil A," http://www.simtd.de/index.dhtml/object.media/deDE/8154/CS/-/backup_publications/Projektergebnisse/simTD-TP5-Abschlussbericht_Teil_A_Manteldokument_V10.pdf, Jun. 2013, accessed on 05.12.2016.

[19] F. Chucholowski, T. Tang, and M. Lienkamp, "Teleoperated Driving Robust and Secure Data Connections," *ATZelektronik worldwide*, vol. 9, no. 1, pp. 42–45, Feb 2014. [Online]. Available: https://doi.org/10.1365/s38314-014-0226-x

[20] J. Davis, C. Smyth, and K. McDowell, "The Effects of Time Lag on Driving Performance and a Possible Mitigation," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 590–593, June 2010.

[21] F. E. Chucholowski, "Evaluation of Display Methods for Teleoperation of Road Vehicles," *Journal of Unmanned System Technology*, vol. 3, no. 3, pp. 80–85, 2016.

[22] T. Tang, P. Vetter, M. Lienkamp, S. Finkl, and K. Figel, "Teleoperated Road Vehicles – The "Free Corridor" as a Safety Strategy Approach," in *Mechanical Design and Power Engineering*. Trans Tech Publications, 2014.

[23] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 927–941, Oct 2007.

[24] J.-M. Georg and F. Diermeyer, "An adaptable and immersive real time interface for resolving system limitations of automated vehicles with teleoperation," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Bari, Italy: IEEE, Oct 2019, pp. 2659–2664.

[25] R. Liu, D. Kwak, S. Devarakonda, K. Bekris, and L. Iftode, "Investigating Remote Driving over the LTE Network," in *Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '17, 2017. [Online]. Available: http://doi.acm.org/10.1145/3122986.3123008

[26] S. Vozar and D. M. Tilbury, "Driver modeling for teleoperation with time delay," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3551 – 3556, 2014, 19th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016421555

[27] J.-M. Georg, E. Putz, and F. Diermeyer, "Longtime effects of videoquality, videocanvases and displays on situation awareness during teleoperation of automated vehicles*," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Toronto, ON, Canada: IEEE, Oct 2020, pp. 248–255.

[28] F. Greis, "Per Fernsteuerung durch die Baustelle," https://www.golem.de/news/autonomes-fahren-per-fernsteuerung-durch-die-baustelle-1906-141791.html, Jun. 2019, accessed on 22.11.2020.

[29] S. Neumeier, S. Stapf, and C. Facchi, "The Visual Quality of Teleoperated Driving Scenarios - How good is good enough?" in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. Montreal, Canada: IEEE, Oct 2020, (Accepted Paper).

[30] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3197–3212, 2020.

[31] S. Paris, "A gentle introduction to bilateral filtering and its applications," in *ACM SIGGRAPH 2007 courses*, 2007, pp. 3–es.

[32] khanhlvg, tombstone, a googler, srjoglekar246, and pkulzc, "TensorFlow 2 Detection Model Zoo," https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md, 2020, Accessed: 10.11.2020.

[33] Y.-C. Su and K. Grauman, "Learning compressible 360° video isomers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[34] A. Schimpe, S. Hoffmann, and F. Diermeyer, "Adaptive video configuration and bitrate allocation for teleoperated vehicles," 2021.

[35] "Waymo open dataset: An autonomous driving dataset," 2019.

[36] Google, "Live-Encoder-Setting, Bitrates and Resulotion (org: Live-Encoder-Einstellungen, Bitraten und Aufloesungen)," https://support.google.com/youtube/answer/2853702, Accessed on 11.10.2020.

[37] Adobe Developer Connection, "Recommended bit rates for live streaming," https://www.adobe.com/devnet/adobe-media-server/articles/dynstream_live/popup.html, Accessed on 11.10.2018.

[38] M. Inc, "Command Line Options - x265 documentation)," https://x265.readthedocs.io/en/stable/cli.html, Accessed on 11.10.2020.

[39] Dark_Shikari, "x264 FFmpeg Options Guide - Linux Encoding)," https://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping, Accessed on 24.07.2021.

[40] OpenCV, "Image Filtering - Image Processing," https://docs.opencv.org/4.2.0/d4/d86/group__imgproc__filter.html, 2020, Accessed: 10.11.2020.

[41] K. Hinum, "NVIDIA GeForce RTX 2070 (Desktop) GPU - Benchmarks and Specs," https://www.notebookcheck.net/NVIDIA-GeForce-RTX-2070-Desktop-GPU-Benchmarks-and-Specs.399491.0.html, Jan. 2019, accessed on 06.01.2022. [Online]. Available: https://www.notebookcheck.net/NVIDIA-GeForce-RTX-2070-Desktop-GPU-Benchmarks-and-Specs.399491.0.html

[42] O. team, "OpenCL - OpenCV," https://opencv.org/opencl/, Apr. 2019, accessed on 06.01.2022. [Online]. Available: https://opencv.org/opencl/

[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[44] D. J. Leiner, "Our research's breadth lives on convenience samples a case study of the online respondent pool "sosci panel"," *SCM Studies in Communication and Media*, vol. 5, no. 4, pp. 367–396, 2017.

[45] A. Ostaszewska and S. Żebrowska-Łucyk, "The method of increasing the accuracy of mean opinion score estimation in subjective quality evaluation," in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*. Springer, 2010, pp. 315–329.

[46] I. T. Union, "P.913: Methods for the subjective assessment of video quality, audio quality and audiovisual quality of internet video and distribution quality television in any environment," https://www.itu.int/rec/T-REC-P.913, mar 2016, Accessed: 03.07.2019.

[47] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

[48] SurveyCircle, "Research website surveycircle. published 2016," https://www.surveycircle.com, 2021, accessed on 08.05.2021.

[49] THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)," http://data.europa.eu/eli/reg/2016/679/oj, 2021, accessed on 30.05.2021.

[50] C. Croux and C. Dehon, "Influence functions of the spearman and kendall correlation measures," *Statistical methods & applications*, vol. 19, no. 4, pp. 497–515, 2010.

[51] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[52] A. K. Akan, M. A. Genc, and F. T. Y. Vural, "Just noticeable difference for machines to generate adversarial images," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1901–1905.

[53] S. Neumeier and C. Facchi, "Towards a Driver Support System for Teleoperated Driving," in *22nd Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct 2019.

[54] NVIDIA Corporation, "Nvidia drive agx developer kit," https://developer.nvidia.com/drive/drive-agx, 2021, accessed on 08.05.2021.

[55] S. Oza and K. R. Joshi, "Cuda based fast bilateral filter for medical imaging," in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, ser. 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Feb 2018, pp. 930–935.

[56] J. M. Georg, J. Feiler, S. Hoffmann, and F. Diermeyer, "Sensor and actuator latency during teleoperation of automated vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA: IEEE, Nov. 2020, pp. 760–766.

[57] K. Doki, K. Suzuki, A. Torii, S. Mototani, Y. Funabora, and S. Doki, "AR video presentation using 3D LiDAR information for operator support in mobile robot teleoperation," in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. Herl'any, Slovakia: IEEE, Jan. 2021, pp. 000 059–000 064.

Stefan Neumeier is a PhD student in the BayWISS-Verbundkolleg Mobilitaet & Verkehr with C-ECOS at Technische Hochschule Ingolstadt and with the chair of Connected Mobility at Technische Universität München. His research interest is on Teleoperated Driving with a focus on how to enable reliable remote control of vehicles in everyday traffic scenarios using the cellular network.

Stefan Neumeier received his B. Sc. in Computer Science at Technische Hochschule Ingolstadt in 2014. He also received his M. Sc. in Applied Research in Engineering Sciences with specialization in Computer Science at Technische Hochschule Ingolstadt in 2016.

Vaibhav Bajpai is an independent research group leader at CISPA Helmholtz Center for Information Security, Hannover. Previously, he was a senior researcher at the Department of Computer Science at the Technische Universität München. He received his PhD (2016) and Masters (2012) degrees from Jacobs University Bremen. He is the recipient of the best of CCR award (2019), ACM SIGCOMM best paper award (2018), and IEEE CNOM best dissertation award (2017). He is also the recipient of the Preis für die beste Lehre (2020) awarded by the Department of Computer Science at TUM. His current research focuses on improving Internet operations (e.g., performance, security, and privacy) using data-intensive methods and by building real-world systems and models.

Marion Neumeier is a PhD student at the CARISSMA Institute of Automated Driving of the Technische Hochschule Ingolstadt funded by AUDI AG. Her research interest is on Machine Learning with focus on interpretability and its application in the task of vehicle trajectory prediction.

Marion Neumeier received her bachelor's degree in Mechatronics (B. Eng.) at Technische Hochschule Ingolstadt in 2018. She received her master's degree in Automated Driving and Vehicle Safety (M.Eng.) at Technische Hochschule Ingolstadt in 2020.

Christian Facchi is Professor for SW Engineering and Distributed Applications at Technische Hochschule Ingolstadt since 2004. His research interest is currently Vehicle2X-Communication, RFID, Testing in Industry 4.0 and Software Testing. He is leading several public funded (5.2 Mio €) and founded by Industry (1,6 Mio €) projects. Since 2011 he is head of the THI graduate school. Currently he holds since 2013 a research professorship for embedded and distributed systems. He is a member of the German Council of Science and Humanities (Wissenschaftsrat) since 2020.

At the Technische Universität München he studied Computer Science. He finished his Ph.D "Methodology for formal Specification of the ISO/OSI Basic Reference Model" at the chair of Software & Systems Engineering under supervision of Manfred Broy. Afterwards he has been employed for 9 years by the R&D-Department of Siemens Mobile Phones, where he had several Line management and project management positions. His last position has been "Leader Worldwide Strategy SW Development Environment".

Joerg Ott holds the Chair for Connected Mobility at Technische Universität München in the Faculty of Informatics since August 2015. He is also Adjunct Professor at Aalto University, where he was Professor for Networking Technology with a focus on Protocols, Services, and Software from 2005 until 2015. He is interested in understanding, designing, and building Internet-based (mobile) networked systems and services. His research focus is on network and system architectures, protocol design, and applications for mobile systems.