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

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 the C-ECOS, Technische Hochschule Ingolstadt, 85049 Ingolstadt, Germany, and also with the Chair of Connected Mobility, Technical University of Munich, 80333 Munich, Germany (e-mail: stefan.neumeier@thi.de).

Vaibhav Bajpai is with the CISPA Helmholtz Center for Information Security, 30159 Hannover, Germany (e-mail: bajpai@cispa.de).

Marion Neumeier is with the C-IAD, Technische Hochschule Ingolstadt, 85049 Ingolstadt, Germany (e-mail: marion.neumeier@carissma.eu).

Christian Facchi is with the C-ECOS, Technische Hochschule Ingolstadt, 85049 Ingolstadt, Germany (e-mail: christian.facchi@thi.de).

Joerg Ott is with the Chair of Connected Mobility, Technical University of Munich, 80333 Munich, Germany (e-mail: ott@in.tum.de).

Digital Object Identifier 10.1109/TITS.2022.3171718

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 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 re-transmission 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 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 *driving relevant front-camera* [16] they present two compression setups. The first with a resolution of 640 × 480, CRF 25 and H.264 bandwidth, and the second with 320 × 240, 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 videostream 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 videostream 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 320 × 320 and EfficientDet D7 1536 × 1536 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.

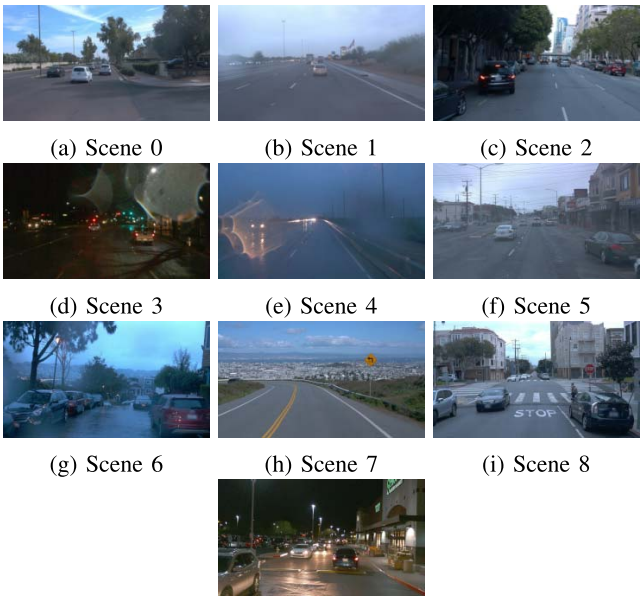
In summary, the following sections present quality perceiving codec-parameters to achieve the lowest bandwidths. This is enhanced by discussing the lane-only approach, where only the lane is kept unblurred, while the rest is blurred. An advancement of this approach is adding important objects, which 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.

[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.

TABLE I

MINIMAL REQUIRED BANDWIDTHS (STUDY COMPRESSION) IN Kbps BASED ON THE RESULTS IN [29], WHERE ONLY ENCODER SETTINGS WERE ADJUSTED

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

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 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_*320 × 320*_coco17_tpu-8* and *efficientdet_d7_coco17_tpu-32*, both received from ModelZoo [32]. FFMpeg is in version (7:4.2.4-1ubuntu0.1).

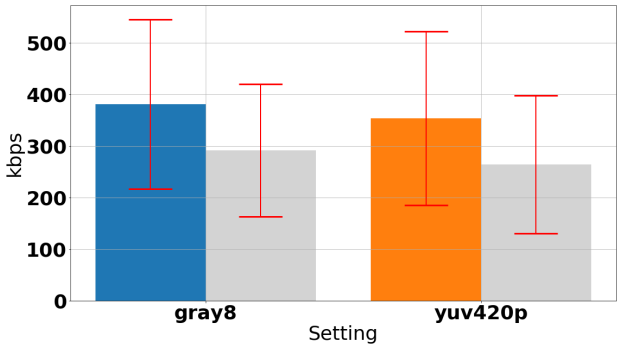


Fig. 2. Comparison of colorspace *gray8* and *yuv420p*. The gray bar indicates the median results for the *remainder* in GBF, while the colored area indicates the median bandwidth required for BF. The red error-bars indicate the standard deviation.

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 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

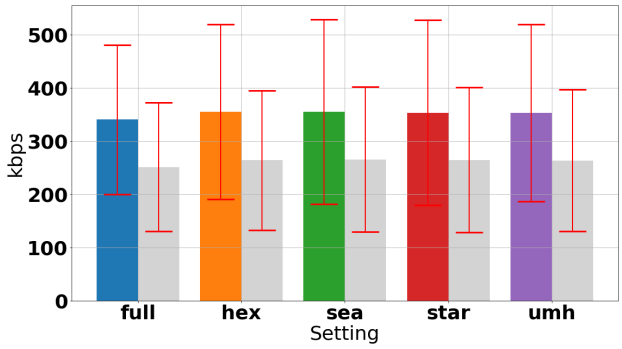


Fig. 3. Comparison of the different motion-estimation search methods on *yuv420p*. The gray bar indicates the median results for the *remainder* in GBF, while the colored area indicates the median bandwidth required for BF. The red error-bars indicate the standard deviation.

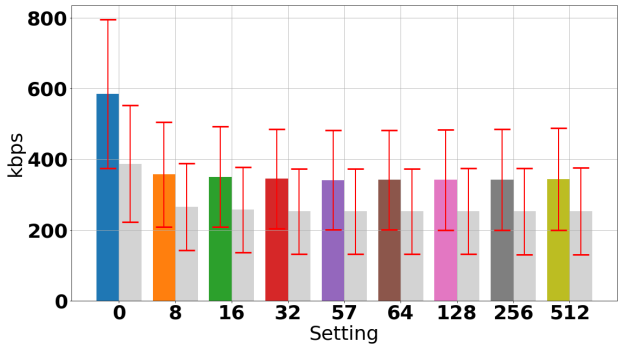


Fig. 4. Comparison of the different motion-estimation search ranges for *yuv420p* and *umh*. The gray bar indicates the median results for the *remainder* in GBF, while the colored area indicates the median bandwidth required for BF. The red error-bars indicate the standard deviation.

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 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 6a) and *mask* (Figure 6b), 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

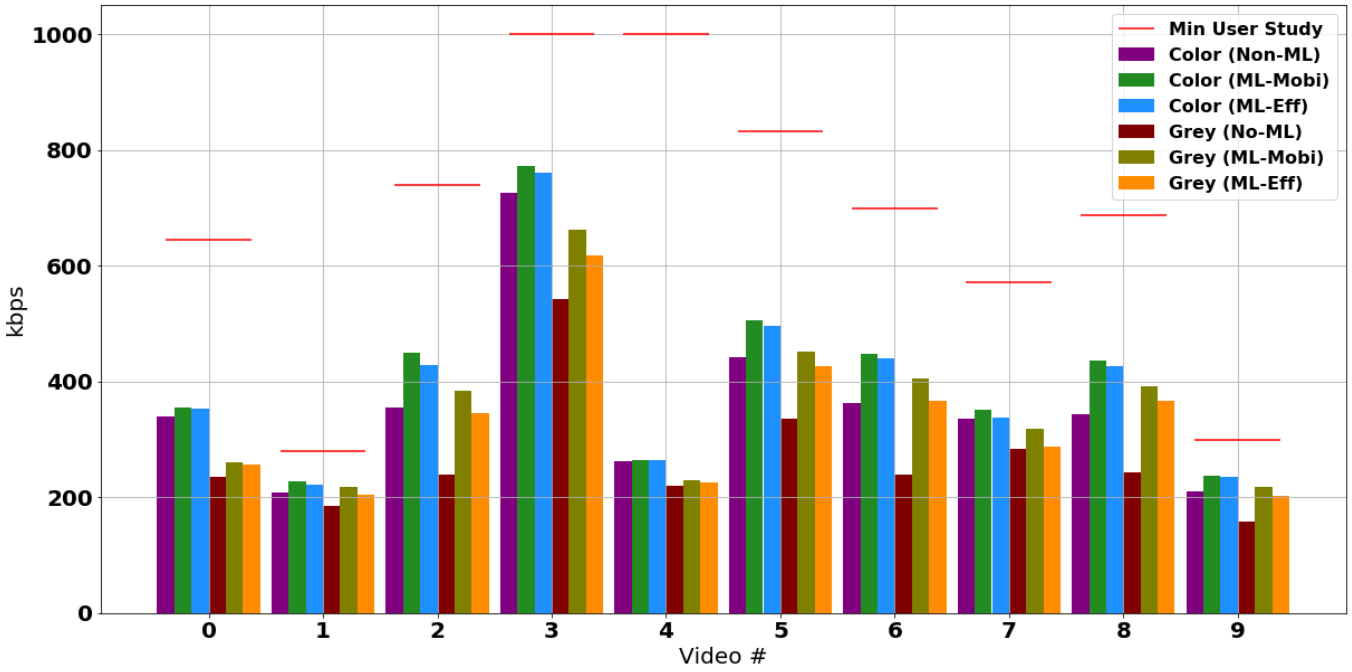


Fig. 5. Comparison of the lane-only and ML results for *yuv420p*, *umh* and a search range of *57*. The red bars indicate the bandwidth-requirements identified by [29] (Table I).

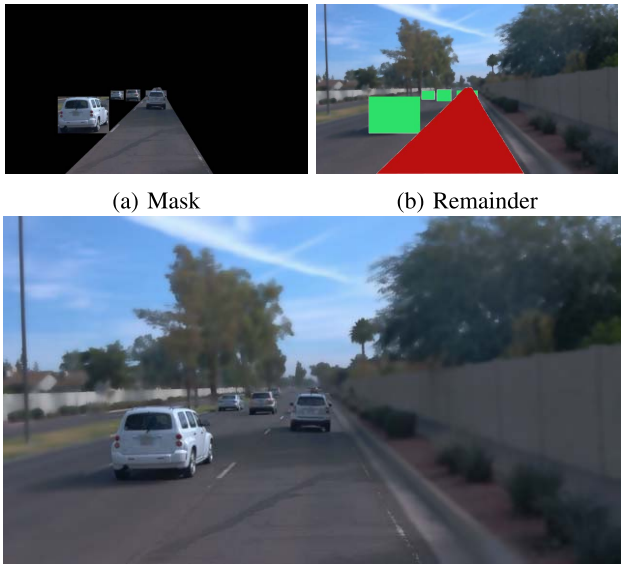
Neumeier *et al.* [29]. It can be seen that the results of the compression methods proposed by this work fall bellow these bandwidth baselines for each video. As such, it can be said, that the approach can help to reduce the bandwidth required for the stream. The average streams are 53% (BF) and 40% (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 1536 × 1536 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 320 × 320 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].



(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.

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.

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

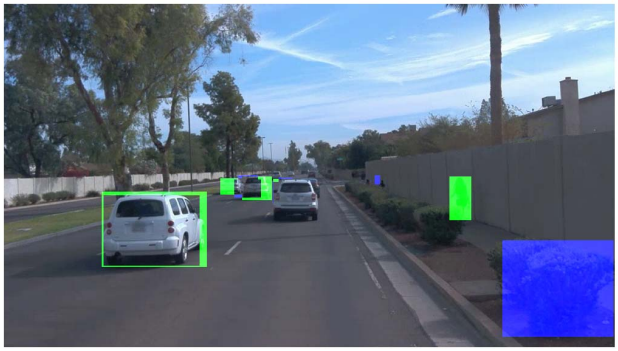


Fig. 7. Difference between the to ML approaches. Blue indicates areas detected by MobilNet only, green indicates areas detected by EfficientDet only.

Fig. 8. Field of view as used in the proposed approach.

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

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

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 320×320*:* The *SSD MobileNet v2* 320×320 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.

2) EfficientDet D7 1536×1536*:* With *EfficientDet D7* 1536×1536 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)

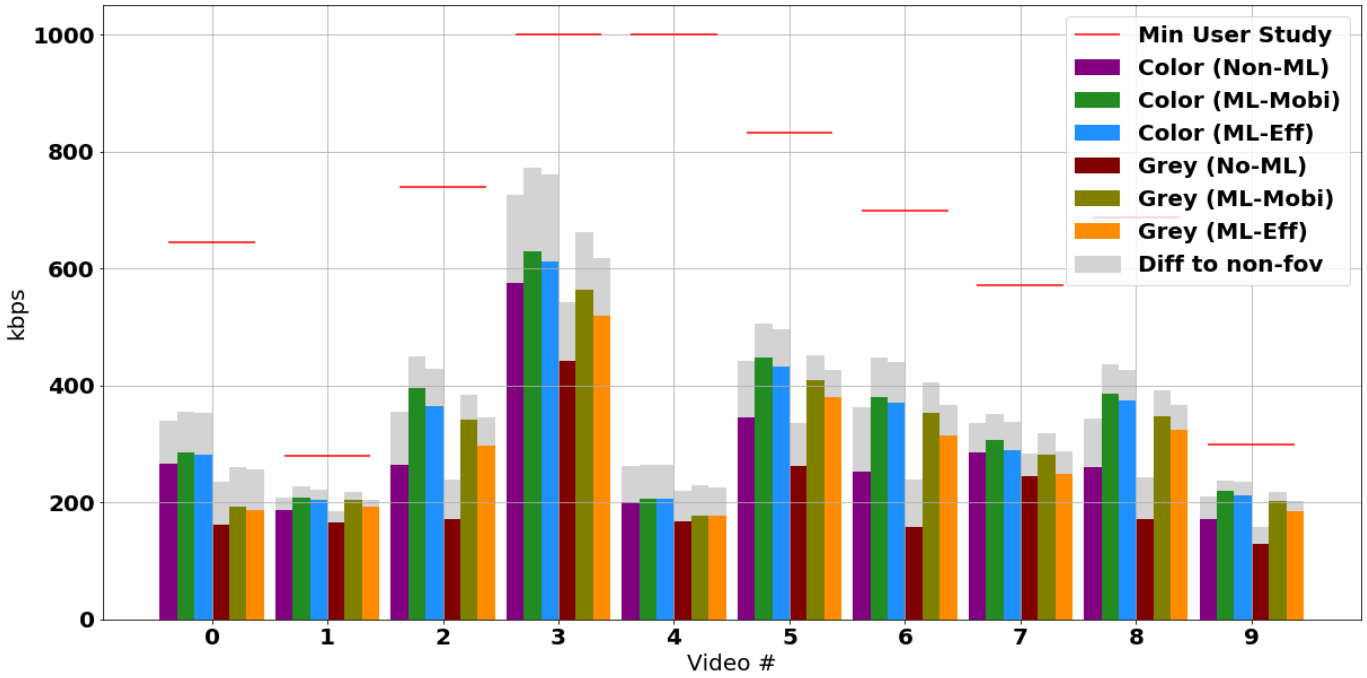


Fig. 9. Comparison of the field of view lane-only and ML results for *yuv420p*, *umh* and a search range of *57*. The red bars indicate the bandwidth requirements identified by [29] (Table I). The gray bars show the difference between the non-fov (gray) and the fov (color) approach.

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 320 × 320*:* 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 1536 × 1536*:* 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 8s–10s 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 1600×900, 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 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 $t > 250$ s. As the participants were presented 20 videos with about a length of 10 s each, the remaining time for reading the introduction and rating the video would be 50 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 s, with a range of 252 s to 1486 s.

TABLE III
SCENARIO-BASED AVERAGE MOS FOR ALL VIDEO CLIPS WITH *Study Compression*

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

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 (∼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 (∼61%) were ranked as driveable for the *basic compression*, while 22 combinations (∼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

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

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

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 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.

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

[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.

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

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

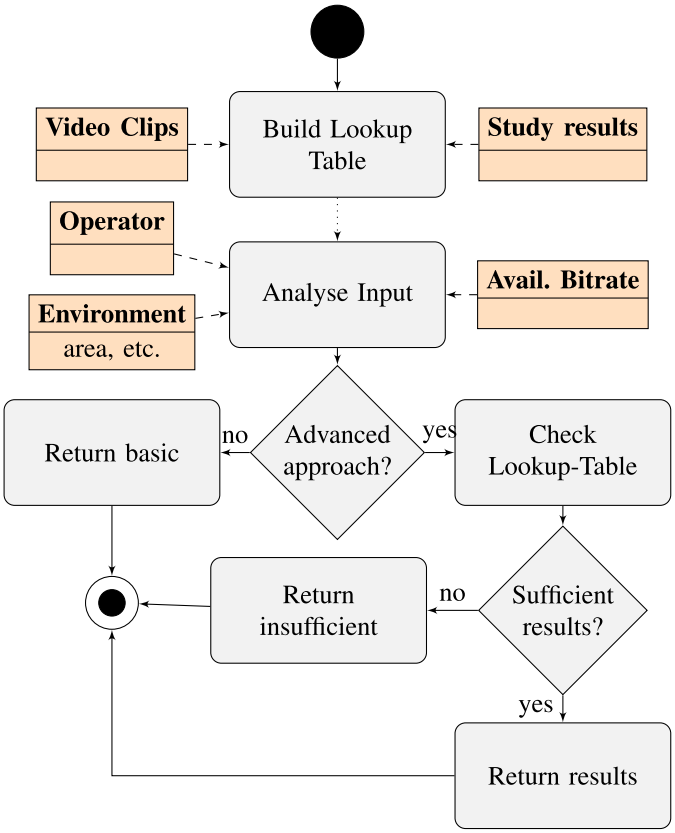


Fig. 10. Flow diagram of the algorithm from building the look-up table and generating the proposed combinations based on the given environmental conditions and the available bitrate. The process usually starts at the node *Analyse Input* if a lookup-table was defined priorly.

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 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: 1280×720, 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

(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.

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

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

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.

REFERENCES

[1] T. Litman, "Autonomous vehicle implementation predictions—Implications for transport planning," Victoria Transp. Policy Inst., Victoria, BC, Canada, Tech. Rep., Mar. 2019.

[2] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE Int., On-Road Automated Driving (ORAD) Committee, Warrendale, PA, USA, Jun. 2018.

[3] A.-K. Frison *et al.*, "In UX we trust: Investigation of aesthetics and usability of driver-vehicle interfaces and their impact on the perception of automated driving," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, May 2019, pp. 144:1–144:13, doi: 10.1145/3290605.3300374.

[4] T. W. Victor, E. Tivesten, P. Gustavsson, J. Johansson, F. Sangberg, and M. L. Aust, "Automation expectation mismatch: Incorrect prediction despite eyes on threat and hands on wheel," *Hum. 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 *Proc. 19th Int. Workshop Mobile Comput. Syst. Appl.*, Feb. 2018, pp. 19–24. [Online]. Available: http://doi.acm.org/10.1145/3177102.3177104

[6] M. Harris. (Jan. 2018). *CES 2018: Phantom Auto Demonstrates First Remote-Controlled Car on Public Roads*. Accessed: Nov. 28, 2018. [Online]. Available: https://spectrum.ieee.org/cars-that-think/transportation/self-driving/ces-2018-phantom-auto-demonstrates-first-remotecontrolled-car-on-public-roads,

[7] *Tele-Operated Driving (ToD): System Requirements Analysis and Architecture*, 5GAA Automot. Assoc., Munich, Germany, Sep. 2021.

[8] A. Davies. (May 2017). *Nissan's Path to Self-Driving Cars? Humans in Call Centers*. Accessed: Oct. 21, 2018. [Online]. Available: https://www.wired.com/2017/01/nissans-self-driving-teleoperation/

[9] Ericsson. (Jun. 2017). *Remote Operation of Vehicles With 5G*. Accessed: Mar. 10, 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. (Mar. 2019). *The War to Remotely Control Self-Driving Cars Heats Up*. Accessed: Apr. 4, 2019. [Online]. Available: https://www.wired.com/story/designated-driver-teleoperations-self-driving-cars/

[11] S. Neumeier, N. Gay, C. Dannheim, and C. Facchi, "On the way to autonomous vehicles teleoperated driving," in *Proc. Automot. Meets Electron., 9th GMM-Symp. (AmE)*. Dortmund, Germany: VDE, 2018, pp. 1–6.

[12] S. Neumeier, E. A. Walelgne, V. Bajpai, J. Ott, and C. Facchi, "Measuring the feasibility of teleoperated driving in mobile networks," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2019, pp. 113–120.

[13] K. Schilling, H. Roth, and R. Lieb, "Teleoperations of rovers. From Mars to education," in *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 1, Jul. 1997, pp. SS257–SS262.

[14] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[15] A. F. Winfield, "Future directions in tele-operated robotics," in *Telerobotic Applications*. Professional Engineering Publishing: London, U.K., 2000.

[16] S. Gnatzig, F. Chucholowski, T. Tang, and M. Lienkamp, "A system design for teleoperated road vehicles," in *Proc. 10th Int. Conf. Informat. Control, Autom. Robot.*, Jul. 2013, pp. 231–238.

[17] J. Wannstrom, "Lte-advanced," *3rd Gener. Partnership Project (3GPP)*, 2012.

[18] Sichere Intelligente Mobilitaet Testfeld Deutschland. (Jun. 2013). *Deliverable D5.5—Teil A TP5-Abschlussberich—Teil A*. Accessed: Dec. 5, 2016. [Online]. Available: http://www.simtd.de/index.dhtml/object.media/deDE/8154/CS/-/backup_publications/Projektergebnisse/simTD-TP5-Abschlussbericht_Teil_A_Manteldokument_V10.pdf

[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, doi: 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 Trans. Robot.*, vol. 26, no. 3, pp. 590–593, Jun. 2010.

[21] F. E. Chucholowski, "Evaluation of display methods for teleoperation of road vehicles," *J. Unmanned Syst. Technol.*, vol. 3, no. 3, pp. 80–85, Feb. 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*. Freienbach, Switzerland: Trans Tech Publications, 2014.

[23] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological interfaces for improving mobile robot teleoperation," *IEEE Trans. Robot.*, 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 *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Bari, Italy, 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 *Proc. 9th Int. Conf. Automot. User Interfaces Interact. Veh. Appl. (AutomotiveUI)*, Sep. 2017, doi: 10.1145/3122986.3123008.

[26] S. Vozar and D. M. Tilbury, "Driver modeling for teleoperation with time delay," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 3551–3556, 2014. [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 *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Toronto, ON, Canada, Oct. 2020, pp. 248–255.

[28] F. Greis. (Jun. 2019). *Per Fernsteuerung Durch die Baustelle*. Accessed: Nov. 22, 2020. [Online]. Available: https://www.golem.de/news/autonomes-fahren-per-fernsteuerung-durch-die-baustelle-1906-141791.html

[29] S. Neumeier, S. Stapf, and C. Facchi, "The visual quality of teleoperated driving scenarios how good is good enough?" in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Montreal, QC, Canada, Oct. 2020, pp. 1–8.

[30] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, "Deep learning-based gait recognition using smartphones in the wild," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3197–3212, 2020.

[31] S. Paris, "A gentle introduction to bilateral filtering and its applications," in *Proc. ACM SIGGRAPH*, 2007, p. 3.

[32] (2020). *Khanhlvg, Tombstone, a Googler, Srjoglekar246, and Pkulzc, TensorFlow 2 Detection Model Zoo*. Accessed: Nov. 10, 2020. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

[33] Y.-C. Su and K. Grauman, "Learning compressible 360° video isomers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7824–7833.

[34] A. Schimpe, S. Hoffmann, and F. Diermeyer, "Adaptive video configuration and bitrate allocation for teleoperated vehicles," in *Proc. IEEE Intell. Vehicles Symp. Workshops (IV Workshops)*, Nagoya, Japan, Jul. 2021, pp. 11–17.

[35] *Waymo Open Dataset: An Autonomous Driving Dataset*. Mountain View, CA, USA: Waymo, 2019.

[36] Google. *Live-Encoder-Setting, Bitrates and Resulotion (Org: Live-Encoder-Einstellungen, Bitraten und Aufloesungen)*. Accessed: Oct. 11, 2020. [Online]. Available: https://support.google.com/youtube/answer/2853702

[37] Adobe Developer Connection. *Recommended Bit Rates for Live Streaming*. Accessed: Oct. 11, 2018. [Online]. Available: https://www.adobe.com/devnet/adobe-media-server/articles/dynstream_live/popup.html

[38] MulticoreWare. *Command Line Options—X265 Documentation*. Accessed: Oct. 11, 2020. [Online]. Available: https://x265.readthedocs.io/en/stable/cli.html

[39] Dark_Shikari. *X264 FFmpeg Options Guide—Linux Encoding*. Accessed: Jul. 24, 2021. [Online]. Available: https://sites.google.com/site/linuxencoding/x264-ffmpeg-mapping

[40] OpenCV. (2020). *Image Filtering—Image Processing*. Accessed: Nov. 10, 2020. [Online]. Available: https://docs.opencv.org/4.2.0/d4/d86/group__imgproc__filter.html

[41] K. Hinum. (Jan. 2019). *NVIDIA GeForce RTX 2070 (Desktop) GPU—Benchmarks and Specs*. Accessed: Jan. 6, 2022. [Online]. Available: https://www.notebookcheck.net/NVIDIA-GeForce-RTX-2070-Desktop-GPU-Benchmarks-and-Specs.399491.0.html

[42] OpenCV Team. (Apr. 2019). *OpenCL—OpenCV*. [Online]. Available: https://opencv.org/opencl/

[43] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 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,'" *Stud. Commun. Media*, vol. 5, no. 4, pp. 367–396, 2016.

[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*. Germany, Berlin: Springer, 2010, pp. 315–329.

[46] ITU-T Union. (Mar. 2016). *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*. Accessed: Jul. 3, 2019. [Online]. Available: https://www.itu.int/rec/T-REC-P.913

[47] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (MOS) revisited: Methods and applications, limitations and alternatives," *Multimedia Syst.*, vol. 22, no. 2, pp. 213–227, Mar. 2016.

[48] SurveyCircle. (2021). *Research Website Surveycircle. Published 2016*. Accessed: May 8, 2021. [Online]. Available: https://www.surveycircle.com

[49] (2021). *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)*. Accessed: May 30, 2021. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj

[50] C. Croux and C. Dehon, "Influence functions of the Spearman and Kendall correlation measures," *Stat. Methods Appl.*, vol. 19, no. 4, pp. 497–515, Nov. 2010.

[51] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, 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 *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1901–1905.

[53] S. Neumeier and C. Facchi, "Towards a driver support system for teleoperated driving," in *Proc. 22nd IEEE Intell. Transp. Syst. Conf. (ITSC)*, Auckland, New Zealand, Oct. 2019, pp. 4190–4196.

[54] NVIDIA Corporation. (2021). *NVIDIA Drive AGX Developer Kit*. Accessed: May 8, 2021. [Online]. Available: https://developer.nvidia.com/drive/drive-agx

[55] S. Oza and K. R. Joshi, "CUDA based fast bilateral filter for medical imaging," in *Proc. 5th Int. Conf. Signal Process. Integr. Netw. (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 *Proc. IEEE Intell. Vehicles Symp. (IV)*, Las Vegas, NV, USA, 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 *Proc. IEEE 19th World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Herl'any, Slovakia, Jan. 2021, pp. 59–64.

Stefan Neumeier received the B.Sc. degree in computer science and the M.Sc. degree in applied research in engineering sciences with specialization in computer science from the Technische Hochschule Ingolstadt in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the BayWISS-Verbundkolleg Mobilitaet & Verkehr, C-ECOS, Technische Hochschule Ingolstadt, and with the Chair of Connected Mobility, Technische Universität München.

His research interest includes teleoperated driving with a focus on how to enable reliable remote control of vehicles in everyday traffic scenarios using the cellular networks.

Vaibhav Bajpai received the master's and Ph.D. degrees from Jacobs University Bremen in 2012 and 2016, respectively. He is currently an Independent Research Group Leader at the CISPA Helmholtz Center for Information Security, Hannover. Previously, he was a Senior Researcher at the Department of Computer Science, Technische Universität München. 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. He was a recipient of the Best of CCR Award (2019), ACM SIGCOMM Best Paper Award (2018), and IEEE CNOM Best Dissertation Award (2017). He was a recipient of the Preis für die Beste Lehre (2020) awarded by the Department of Computer Science, TUM.

Marion Neumeier received the bachelor's degree (B.Eng.) in mechatronics and the master's degree (M.Eng.) in automated driving and vehicle safety from the Technische Hochschule Ingolstadt in 2018 and 2020, respectively, where she is currently pursuing the Ph.D. degree with the CARISSMA Institute of Automated Driving, funded by AUDI AG.

Her research interest includes machine learning with focus on interpretability and its application in the task of vehicle trajectory prediction.

Christian Facchi received the Ph.D. degree in methodology for formal specification of the ISO/OSI basic reference model from the Chair of Software & Systems Engineering under supervision of Manfred Broy.

At the Technische Universität München, he studied computer science. Afterwards, he has been employed for nine years by the Research and Development Department of Siemens Mobile Phones, where he had several line management and project management positions. His last position has been the Leader of the Worldwide Strategy SW Development Environment. He has been a Professor of SW engineering and distributed applications at Technische Hochschule Ingolstadt since 2004. Since 2011, he has been the Head of the THI Graduate School. Since 2013, he holds a Research Professorship for embedded and distributed systems. He is leading several public funded (5.2 Mio €) and founded by industry (1,6 Mio €) projects. His research interests include vehicle2x-communication, RFID, testing in industry 4.0, and software testing. He has been a member of the German Council of Science and Humanities (Wissenschaftsrat) since 2020.

Joerg Ott has been the Chair for Connected Mobility, Faculty of Informatics, Technische Universität München, since August 2015. He is currently an Adjunct Professor at Aalto University, where he was a Professor of networking technology with a focus on protocols, services, and software, from 2005 to 2015. He is interested in understanding, designing, and building internet-based (mobile) networked systems and services. His research interests include network and system architectures, protocol design, and applications for mobile systems.