ECE 445

SENIOR DESIGN LABORATORY

PROJECT PROPOSAL

Remote Driving System

Team #17

BO PANG (bopang5@illinois.edu)
JIAHAO WEI (jiahaow4@illinois.edu)
KANGYU ZHU (kangyuz2@illinois.edu)

TA: Yi Wang

March 11, 2023

Contents

1 Introduction

1.1 Problem

Traditional chauffeuring service provides people with convenient and safe transportation solution. However, it also has many disadvantages. First, the chauffeur might experience distraction within the car, which might raise safety issue. Also, it lacks flexibility and efficiency since it requires the physical presence of the chauffeur. This inefficiency causes a high cost to cover driver compensation and overtime[1].

With the help of a remote driving system, the chauffeur can operate the owner's automobile remotely. This solution offers more flexibility in terms of scheduling, as remote chauffeurs can operate the vehicle from anywhere at any time. It also does not require the physical presence of the chauffeur, which can result in lower labor costs[2]. Furthermore, it can provide an additional layer of safety as the chauffeur is not physically in the vehicle, reducing the risk of distraction or driver error.

1.2 Solution

Our remote driving system will provide real-time feedback of the car's external environment and information (e.g., car's current speed) to the remote chauffeurs. Through the use of advanced technologies, these chauffeurs can remotely operate the car's movement using the designated devices.

This system consists of four subsystems. The MR subsystem displays the information including car's external environment and status sent back by the automobile control subsystem. The driver control subsystem records commands of the chauffeur, which will be sent back to the automobile. The server subsystem help transmit message sent between these subsystems, achieving the result of remote driving.

1.3 Visual Aid

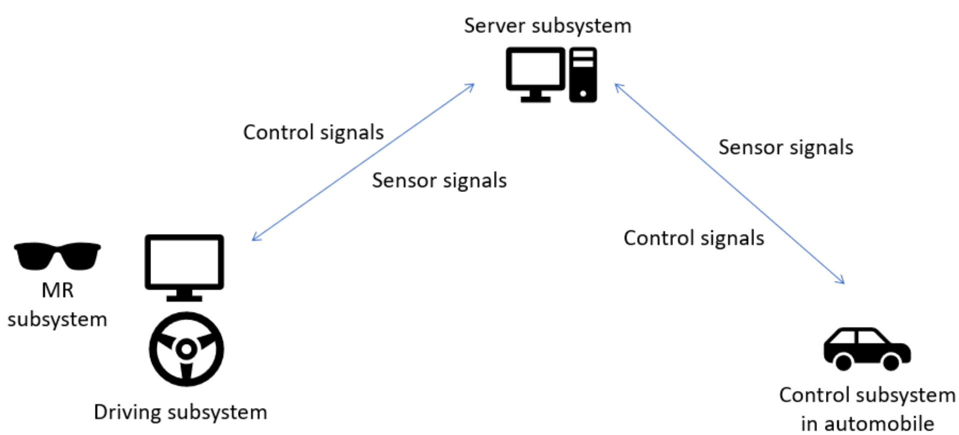

Figure 1: Visual Aid for Remote Driving System

1.4 High-level requirements list

We will substitute a toy car (TurtleBot3) for the automobile to reduce the complexity of the control subsystem.

- Under a chauffeur's operation, the toy car (TurtleBot3) should be to navigate through the whole campus.
- The delay of the video sent back from the toy car (TurtleBot3) should be at least within 500 ms. The delay of the driver's command should be within 100 ms.
- The server can handle at least two connections from different chauffeurs. All chauffeurs and view the status of the toy car (TurtleBot3), and the one who will drive the toy car will be selected according to a policy.

2 Design

2.1 Block Diagram

Our entire system consists of a PC server and three subsystems. The PC server receives, processes, and sends information to interact with the three subsystems. In particular, the Driver control subsystem is mainly responsible for receiving and sending control commands. The Hololens2 subsystem is responsible for presenting real-time information at the user end. The TurtleBots3 subsystem is capable of performing movement based on instructions and capturing real-time information of its environment.

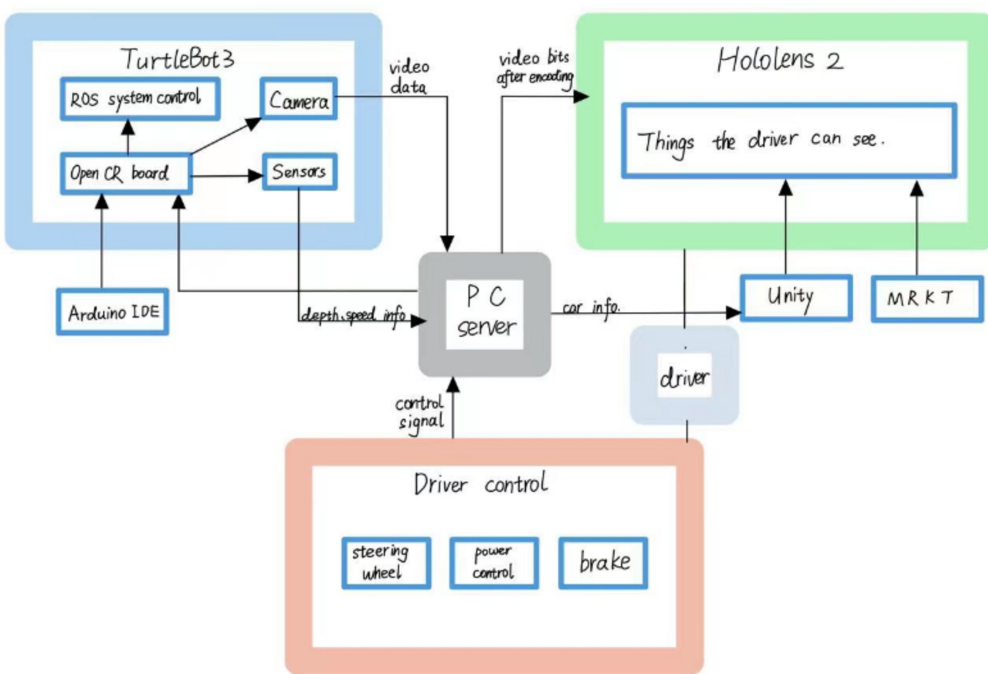

Figure 2: Block Diagram

2.2 System Overview

In this project, we will develop a remote driving system. The driver will wear HoloLens 2 glasses and remotely drive a car using this system. The car's camera will transmit environmental information around the car to the PC, which will synchronize the video information in front of the HoloLens 2 glasses. The driver will send control signals to the car's control board via the driving system on the PC. The car will analyze the control signals and control its movement through the OpenCR control board.

The main platforms we use are Unity, Arduino IDE, and the OpenCR development board. Unity3D is a comprehensive game development tool developed by Unity Technologies, allowing you to easily create interactive content such as 3D video games, architectural visualizations, and real-time 3D animations on multiple platforms. Through Unity, vehicle driving information can be displayed on Hololens 2, increasing driving safety. Arduino IDE is used to control the OpenCR development board, which controls the hardware operation of OpenCR. The OpenCR is the main controller board for TurtleBot3 and is designed for embedded ROS system development. The motion of the vehicle is controlled through the OpenCR development board.

2.3 Subsystem Requirements

2.3.1 TurtleBot3

The TurtleBot3 system needs to transmit video to the server. As driving requires low latency video, the UDP protocol is more suitable for our project than TCP. We plan to use Socket and UDP protocols for real-time video transmission and currently plan to use a resolution of 720p. We will use Socket and TCP to establish reliable connections between the server and client. We will use OpenCV or EmguCV library to capture camera images. Due to the 64kbit size limit of UDP packets, we plan to compress the images using JPEG and send them using UDP protocol at the end.

Connection between TurtleBots3 modules and other sub-modules: The 360-degree camera on TurtleBots3 captures environmental information, which is then transmitted to the server via the OpenCR development board. The sound collector on TurtleBots3 captures surrounding audio information, which is transmitted to the server to facilitate communication between the driver and passengers. Meanwhile, TurtleBots3 also receives control signals from the server, which are used to control the vehicle's movement via the OpenCR and ROS operating systems.

TurtleBots3 is the most critical component in this project, used to control the vehicle through the OpenCR module on it. Development is done using the Arduino IDE, and remote communication is achieved by controlling information transmission and reception on the OpenCR development board. This part of the project requires us to understand the working principle of microprocessors and to develop their functions. OpenCR is the main controller board for TurtleBot3, designed for embedded systems development in ROS, providing complete open-source hardware and software. By understanding the ROS embedded system and processing the signals from the server, better control of TurtleBots3 can be achieved.

2.3.2 PC Server

The PC server system plays the most important role in information exchange as the bridge for signal control. It needs to decode the video information sent by the TurtleBot3 and project it onto the HoloLens 2. It also needs to send signals from the Driver Control system to the Turtlebots system to control the car.

Connection between PC server module and other sub-modules: The PC server plays a critical role in controlling the system and acts as a bridge for signal transmission. TurtleBots3 transmits environmental signals to the server, which are then projected onto Hololens 2 through the server and Unity platform. Audio signals are transmitted through the server to facilitate communication between the driver and passengers. The PC server also transmits control signals from the driver to the TurtleBots3 control board, enabling control of the vehicle. Transmission between the PC server and TurtleBots3 is wireless, while PC Server will connect with the driver and Hololens 2 modules use wired connections.

The role of the PC server in the project is to achieve real-time transmission of information security. It facilitates information sharing between the TurtleBots3 module and the driver control module, allowing the driver to have access to the latest environmental information in real-time. Additionally, it enables real-time communication between the driver and passengers. The PC server is an essential factor in ensuring the success of wireless driving. Furthermore, the PC server is connected to multiple platforms. It projects video signals onto the Hololens 2 module through its connection with the Unity platform. It controls the movement of the TurtleBots3 through its transmission with OpenCR and signal analysis using ROS.

2.3.3 MR System

The HoloLens 2 system carries all the learning that the driver can obtain. The Server will project the transmitted video signal onto the HoloLens 2, allowing the driver to see the environment around the car from multiple angles. Through the Unity platform, the running information of the car can be projected onto the HoloLens 2 for the driver's reference.

Connection between the Hololens 2 module and other sub-modules: The Hololens 2 module projects the video through the Unity platform, which is connected to the PC server. Language communication and environment sound sharing are achieved through the audio transmitted by the PC server and the built-in speaker of the Hololens 2. The MR feature of the Hololens 2 allows the driver to see the information around the vehicle more comprehensively and operate the control system in reality. Through MR, the driving process can be made more realistic, creating a more immersive driving experience. The combination of the Hololens 2 and the use of 360-degree video images allows the driver to grasp the information around the TurtleBots3 without being constrained by space, which is more conducive to completing the driving process safely.

The role of Hololens 2 in the project is to provide a more immersive remote driving experience while also making remote driving safer. Because MR video viewing is no longer limited by screen position, drivers can quickly grasp the surrounding environment of the car. At the same time, through the Unity platform, driving information can be presented in front of the driver and provide prompts, allowing the driver to better adjust their driving behavior. Through MR technology, remote driving becomes more convenient and immersive.

2.3.4 Driver Control

Driver Control is the control system for the driver, which should include components such as a steering wheel, accelerator, and brake. The driver sends control information to the PC server by using the corresponding control parts, achieving remote control of the car.

Connection between the Driver Control module and other sub-modules: The driver controls the car by watching the audio-visual feedback projected by the Hololens 2 module. The driver uses the steering wheel, accelerator, and brake pedals in the Driver Control module to control the car. The Driver Control module is directly connected to the PC Server, which receives the control signals from the Driver Control module and transmits them remotely to the car, enabling remote control of the car.

The role of the Driver Control module in the project is to provide a more immersive and realistic remote control experience. By using a driving interface similar to real life, the remote driving process becomes more accessible and safer.

2.4 Tolerance Analysis

2.4.1 UDP wireless transmission

UDP is often used for real-time communication, such as video streaming, because it offers low latency and faster transmission speed compared to TCP. In remote driving scenarios, real-time video communication is necessary for safe and efficient operation. Therefore, we chose to use UDP for video content transmission.

To further improve the transmission speed, we decided to use JPEG image compression for video data. By compressing the images, we can reduce the size of each UDP packet and decrease the transmission delay. However, this approach also introduces some trade-offs, such as potential loss of image quality due to compression and increased processing requirements on both the sender and receiver side.

Overall, the choice of UDP and JPEG compression for video transmission depends on the specific requirements of the application, such as the desired level of image quality and the available network bandwidth.

2.4.2 5G wireless transmission

In our system, the communication between the car and server will connect using WiFi. However, there exist cases where the WiFi signal is weak or it is disconnected, which will significantly compromised the driving experience and even cause accidents. Nowadays fifth generation (5G) mobile plays an important role in providing fast and stable communication[3]. Thus, we try to build a failover mechanism that can switch to 5G signal to ensure continuous communication.

The process of switching to 5G signal is triggered when the WiFi signal quality drops below a certain threshold or when the WiFi connection is lost completely. Once the system detects a connectivity issue, it automatically switches to 4G signal and establishes a new connection with the server through the cellular network.

To ensure a seamless transition between WiFi and 5G, the system will maintain a backup connection over 5G at all times, and switches back to WiFi as soon as the signal strength improves. Additionally, the system will include a mechanism to alert the user in case of any connectivity issues, allowing them to take necessary actions to ensure the safety of the vehicle.

3 Ethics and Safety

Due to safety considerations, our project involves a variety of potential hazards that may arise during the interaction between humans and machines.

• The mental and physical state of the remote driver

  Remote driving system requires designated drivers to wear virtual reality devices, such as glasses, which often have relatively large sizes and weights compared with the normal devices in our everyday life. These devices could possibly cause distraction to the remote driver and increase their susceptibility to fatigue. Moreover, extended use of virtual reality devices can lead to symptoms such as dizziness, significantly compromising the safety of remote driving. These are all factors that could possibly violate "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices" in IEEE code of ethics[4].. To avoid this from happening, we we highly recommend that designated drivers undergo rigorous screening and training before operating this system. Additionally, they must take sufficient rest after driving for an extended period. These steps are essential to guarantee the safety of both the driver and passengers. We prioritize this issue and will continuously work towards improving the system to reduce potential risks.

• Communication delay, stability and resolution of the transmitted video

  The system use cameras to capture the driving situation and transmit it to a remote driver's location. Through virtual reality technology, the substitute driver can interact with the vehicle in real-time, enabling them to navigate the roads and ensure the safety of passengers. However, this technology is not without its challenges. Accroding to ACM code of ethics: ”the emergent properties of systems should be carefully analyzed[5].” In our system, the transmission of the live video feed from the car to the remote location may be unstable and delayed, leading to potential safety risks. Furthermore, areas with weak signals may require a reduction in video resolution, which could further compromise the remote driver's ability to make informed judgments and decisions. To address these issues, it is essential to focus on enhancing the hardware of the cameras to improve their maximum resolution. In addition, optimizing the transmission network can help to enhance stability, reduce latency, and minimize delays during transmission. By addressing these challenges, we can improve the effectiveness and safety of driver substitution, ensuring a seamless and secure driving experience for all passengers.

- The accuracy and precision of the control system

  The instructions given by remote driver are transmitted to the car, allowing them to control its movements remotely. However, inaccuracies or delays in the control system can pose serious safety hazards for both the vehicle and its carring passengers. For instance, on narrow or busy roads, even minor inaccuracies in the control instructions could result in severe accidents or collisions. To minimize these risks, it is imperative that the human-machine interaction system has high control precision. The remote driver's controller must be designed to have similar function and precision to that of a physical steering wheel and provide a precise and responsive control experience. This will enable the remote driver to make more accurate and effective driving decisions, especially in critical situations.

References

[1] 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 amp; Applications*, ser. HotMobile '18, Tempe, Arizona, USA: Association for Computing Machinery, 2018, pp. 19–24, ISBN: 9781450356305. DOI: 10.1145/3177102.3177104. [Online]. Available: https://doi.org/10.1145/3177102.3177104.

[2] P. Kallioniemi, A. Burova, J. Mäkelä, *et al.*, "Multimodal warnings in remote operation: The case study on remote driving," *Multimodal Technologies and Interaction*, vol. 5, p. 44, Aug. 2021. DOI: 10.3390/mti5080044.

[3] U. Saeed, J. Hämäläinen, M. Garcia-Lozano, and G. David González, "On the feasibility of remote driving application over dense 5g roadside networks," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, 2019, pp. 271–276. DOI: 10.1109/ISWCS.2019.8877213.

[4] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html.

[5] ACM. "ACM Code of Ethics and Professional Conduct." (2018), [Online]. Available: https://www.acm.org/code-of-ethics.