Using Simulation to Accelerate Development of User Interfaces for Autonomous Vehicles *

Christopher R. Hudson[0000-0002-6216-8234], Ryan Lucius, Robert Gray, Brandon Powell, Matthew Doude, and Daniel W. Carruth[0000-0003-0707-9252]

Mississippi State University, Starkville, MS 39759, USA
{chudson}@cavs.msstate.edu
{ral264, rg1071}@msstate.edu
{bpowell, mdoude, dwc2}@cavs.msstate.edu
http://www.cavs.msstate.edu/

Abstract. This paper outlines work being conducted at the Center for Advanced Vehicular Systems at Mississippi State University using simulated data to enable interface development for real world vehicles when hardware systems are not available. The ANVEL modeling and simulation tool was used to generate simulated video feeds. These videos feeds were provided to an information server, which was written to accept both simulated and real world camera data, and convert that data into an MJPEG over HTTP stream. An interface was written in Unity 2017 using simulated camera data received as an MJPEG from the information server. This user interface was then tested on a real world autonomous vehicle project, by passing camera data to the information server for conversion to a common vechile interface format as the simulated camera data. This method allowed for the parallel development of user interface, without the need for physical hardware for testing, while maintaining real-time video capability.

Keywords: HCI · Autonomy · Simulation

1 Introduction

Autonomous vehicles present new user interface design challenges not present on traditional vehicles. An autonomous vehicle has several end users who need to understand and interact with different systems on a vehicle. A passenger requires a means to interact with the vehicle to provide goal locations. A researcher working on development of the system will need to visualize and understand the data provided to the system via on board sensors and how the system process that information. A mechanic working to resolve mechanical and electrical issues on the system will need to understand the complex layout of the autonomous vehicle, and which systems are reporting normal data. Autonomous vehicles are a new

* Supported by the Center for Advanced Vehicular Systems, Mississippi State University. NVIDIA hardware was provided through a partnership with NVIDIA Corporation.

technology still being developed. Many research teams operate with a limited number of vehicles shared among several teams working to meet deadlines. Given the need to make progress on various projects in parallel: design, development and deployment of user interfaces are often assigned lower priority for access to the physical hardware. Given these real-world problems, the implementation of these interfaces is often delayed due to the general need for increased access to the hardware. To address the issue of limited access to a shared platform, we propose using simulation to allow user interface designers to access the data they need to evaluate their designs. Many systems on an autonomous vehicle already rely on modeling and simulation of systems to facilitate faster development. Extending this simulation to enable the development and testing of user interfaces has significant benefits.

This paper outlines the use of the Autonomous Navigation Virtual Environment Laboratory (ANVEL), a real-time modeling and simulation tool for ground vehicle robotics, to accelerate the development of interface applications for an autonomous off-road vehicle project known as HALO at the Center for Advanced Vehicular Systems at Mississippi State University [2]. Using ANVEL, we can simulate numerous cameras, LIDAR, IMU, and GPS sensor data in real time. ANVEL provides APIs to access and pass real time data to external user interfaces and devices. To demonstrate this capability, we have implemented a Unity-based interface on an Android tablet. Simulated data is generated by ANVEL and provided to the information server in the same format as the real-world vehicle. The information server then processes and provides reduced vehicle data to the Android tablet for display. Because data coming from the simulator is processed and provided to the information server in the same format as data from the autonomous vehicle, we can directly transfer the interface from simulation to vehicle. This allows for a hardware in the loop capability. Because the tablet can leverage both real world and simulated data, designers are able to rapidly iterate on the interface using simulated data streams and test with vehicle hardware when it is available. By leveraging simulation techniques used by other development processes in autonomous vehicles, the quality of the interface can be improved while reducing development time and resource requirements.

2 Related Work

This systems mimics a hardware in the loop simulation. This method allows the interface device (Android Tablet) to use data generated in a simulation for development and testing, in replacement of data coming from physical cameras on the vehicle, since the vehicle is not available for the interface development team. Hardware in the loop simulation is a common method, used to provide unit testing for physical devices under specific circumstances [3]. It is known that hardware systems can often be unavailable due to time requirements for the development of different systems. Using model based system design helps developers of hardware systems implement different systems which can be tested in simulation at any level [5]. The notion of designing user interface for these same hardware systems can be included in this design paradigm. By allowing for the simulation of the hardware systems generating the data that needs to be visualized, model based system engineering can be applied to interface design.

3 Implementation

To enable simultaneous implementation of both user interface elements and physical systems without competing for resources, the HALO project vehicle was leveraged as a platform for this work. By leveraging a system that was undergoing powertrain electrification, the notion of working in parallel on a system that was unavailable for testing could be achieved. This section outlines the different platforms and tools leveraged, as well as the layout and function of the information processing server which provides data to the interface.

3.1 HALO Project

At the CAVS at MSU, the HALO project is converting a factory issued 2014 Subaru Forester into an all-electric vehicle. This electrification will result in the replacement of the powertrain for the vehicle. As a stock vehicle, the Subaru Forester has 24 city and 32 highway MPG with 250hp with a total weight of 3600lbs. After converting the vehicle to all-electric drive, the 90 kWh nickel-manganese-cobalt (NMC) lithium-ion battery pack gives it an estimated range of 240 miles, 536hp and a total weight of 5300lbs. Additionally during the process of reconfiguring the vehicle, 6 cameras were added to the vehicle. Two forward facing cameras (60 degree FOV), a left facing Camera (190 degree FOV), a right facing camera (190 degree FOV) and two rear-facing cameras (120 degree FOV). Additionally three LIDAR units were installed, two forward LIDAR on the bumper angled down towards the drive path of the vehicle and one LIDAR unit on top of the vehicle.

3.2 Nvidia Drive PX2

To enable the development of autonomous system behavior, as well as gain access to the LIDAR and cameras installed on the HALO vehicle, an Nvidia Drive PX2 was leveraged. The Nvidia Drive PX2 is a GPU/CPU supercomputer designed for autonomous vehicles[4]. In order to pass camera data from the Drive PX2 to the information server, Gstreamer and OpenCV software libraries were leveraged. Gstreamer was used to capture the image data from the camera devices. The raw image stream was then compressed and sent over UDP to the information server which received the compressed image data in OpenCV. The OpenCV image was made available as an MJPEG stream over HTTP. By leveraging a simple HTTP server, real time video could be accessed by multiple devices on the same network. This is outlined in Figure 1.

3.3 ANVEL

The Autonomous Navigation Virtual Environment Laboratory (ANVEL) is a real-time physics based simulation environment, designed to model autonomous systems driving through various environments. ANVEL provides several simulated systems such as LIDAR, GPS, IMU as well as calculations of different forces acting on the simulated vehicle. In order to model the vehicle, ANVEL uses of definition files which describe each component of the simulated vehicle (battery, engine, wheels, mass, etc.). The ANVEL API was leveraged to extract the image data from the simulation environment. The data was received on the information server, and made available over HTTP as an MJPEG stream, in the same way the HALO car provided the video data. This is outlined in Figure 1.

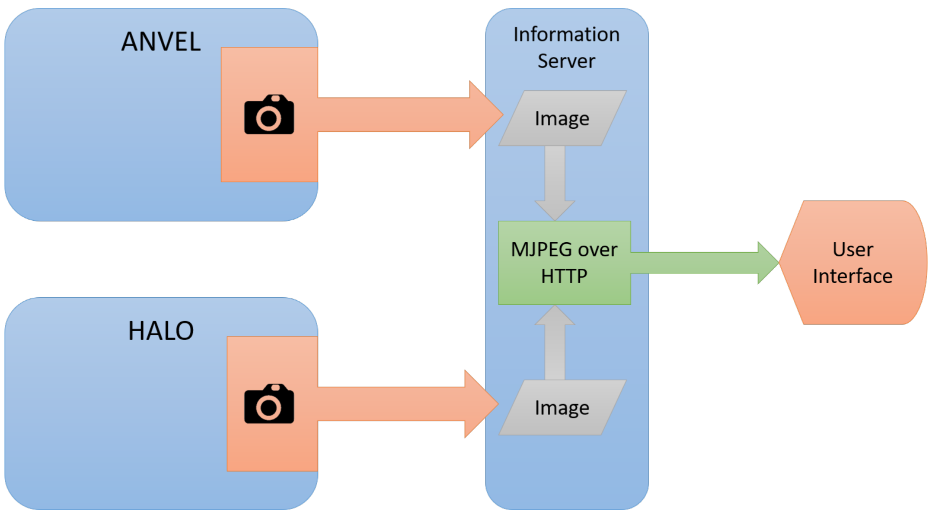

Fig. 1. The flow of video data from source to interface.

3.4 Unity

The user interface was designed using Unity 2017 [1]. Unity was chosen due to its ability to easily deploy to multiple devices such as Android, iOS, and Windows. Additionally, Unity provides scalability in user interfaces to account for a variety of display sizes so that the interface can be deployed on phones, tablets, laptops and desktop systems. This dynamic resolution management combined with ease of deployment made Unity 2017 ideal for implementing the user interface. No plugins were used in this implementation to ensure maximum capability with all systems during deployment.

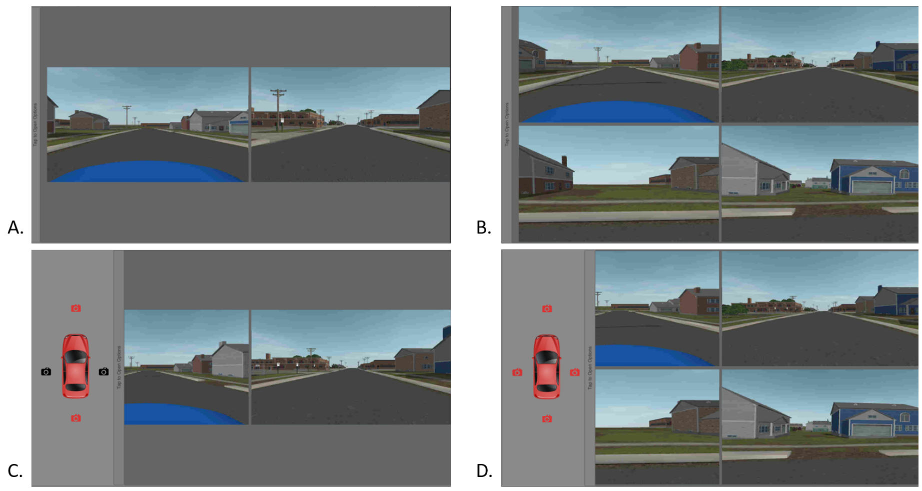

Fig. 2. A) Forward and rear camera on, Full Screen, B) Forward, rear, left, right camera on, Full Screen, C) Forward and rear camera on, selection slider shown, D) Forward, rear, left, right camera on, selection slider shown

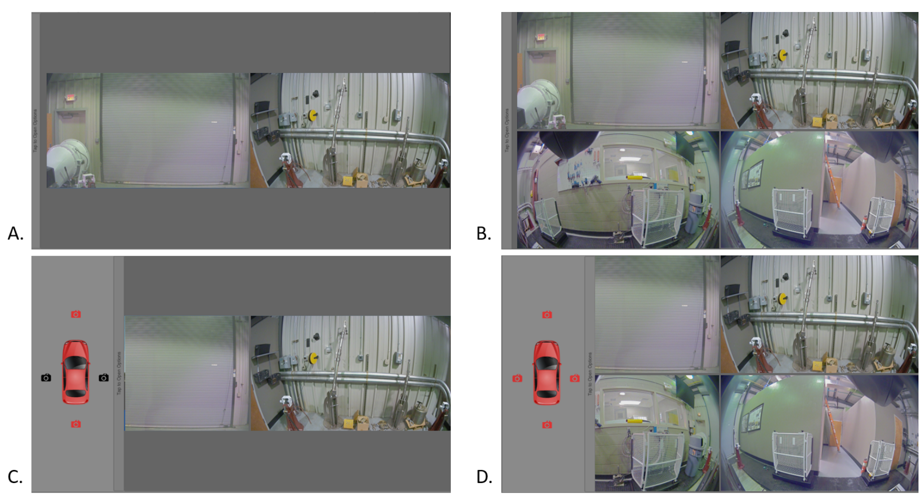

Fig. 3. A) Forward and rear camera on, Full Screen, B) Forward, rear, left, right camera on, Full Screen, C) Forward and rear camera on, selection slider shown, D) Forward, rear, left, right camera on, selection slider shown

4 Results

Video data was successfully integrated into the Unity interface and deployed to phone and tablet devices for monitoring vehicle feeds. The interface software was able to leverage both simulation and real-world data passed through a common information server and provided to the user interface through a common method. Example video feeds from ANVEL can be seen in the user interface in Figure 2. By leveraging MJPEG over HTTP real time performance was achieved.

The capability of the user interface can be seen when connecting to the HALO vehicle. While the simple video monitoring user interface was designed using simulated video feed data, the format of the information was provided to the interface in the same way as real world data. This allowed for the interface to immediately work with the HALO vehicle when it was turned on, without the need for reconfiguration. The integration of video data can been seen in Figure 3.

5 Discussion

Through the use of an information processing server, we have shown that data can be collected from both simulation and real world vehicle and forwarded in a common format to user interfaces. By leveraging simulation, we have shown that tasks such as integration of real-time video displays which rely heavily on the availability of physical hardware in order to test, can leverage simulation in order to speed up the implementation process of user interfaces. The user interface development team was able to develop and iterate on designs while the HALO vehicle was undergoing major system replacement and was unavailable for system testing.

6 Conclusions and Future Work

We have demonstrated use of simulation in order to mimic real world data, specifically leveraging real time video feeds, to design, develop and test user interfaces without the need for physical hardware. By enabling the user interface designer to use simulated data sources that match real-world data sources, work can be conducted in parallel which allows for a more rapid deployment of both physical system, and it's user interface, without having to resolve conflicts in system availability for development. In the future, we hope to expand this work to integrate additional systems present on both the physical and simulated vehicle. Additionally, we would like to explore using the same user interface for multiple vehicles without changing any internal implementation to the user interface, providing only changes to the information processing in the information server, which is vehicle dependent.

References

1. Unity Game Engine. Available from http://www.unity3d.com.
2. Autonomous Navigation Virtual Environment Laboratory (ANVEL). Available from http://anvelsim.com/.
3. Isermann, R., Schaffnit J., & Sinsel, S.: Hardware-in-the-loop simulation for the design and testing of engine-control systems, **7**(5), 643-653 (1999)
4. Nvidia Drive Px2. https://www.nvidia.com/en-us/self-driving-cars/drive-platform/
5. Rousseau, A., Halbach, S., Michaels, L., Shidore, N., Kim, N., Kim, N., Karbowski, D, & Kropinski, M. Electric drive vehicle development and evaluation using system simulation. In: IFAC Proceedings, **47**(3), 7886-7891 (2014)