Software for Real-Time Communication between Multiple HoloLens Devices

Hidde van der Meulen
University College Dublin
Dublin, Ireland
hidde.vandermeulen@ucdconnect.ie

Andrew L. Kun
University of New Hampshire
Durham, NH, United States
andrew.kun@unh.edu

2017 Workshop on Augmented Reality for Intelligent Vehicles.
September 24, 2017, Oldenburg, Germany

Abstract
Sharing information between multiple augmented reality devices is necessary for shared experiences. This work describes a custom tool to share data between multiple HoloLens devices including data from eye-trackers connected to each HoloLens.

Author Keywords
Eye tracking, Augmented Reality, Realtime communication

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

Introduction
Augmented reality glasses such as the Microsoft HoloLens allow the wearer to see additional 3D projections in the real world. The experience is individual and even with a second HoloLens the displayed holograms and interactions are not automatically shared between the devices, leaving the users with disconnected experiences.

The HoloLens keeps track of its rotation and location in a room using outward facing cameras. If a wearer moves through space the hologram presentation is changed accordingly, mimicking real objects. Users can interact with objects using hand gestures and voice commands to reposition, rotate or scale virtual objects.

Hardware

We fitted two Microsoft HoloLenses with Pupil Labs eye-trackers that get a picture of the user's pupils while wearing the HoloLens. Using a USB cable, the eye-trackers are connected to laptops with the Pupil Labs software. Note that HoloLens keeps track of its location and rotation relative to the environment but lacks the ability to capture eye gaze data. In preliminary experiments we found indications that the addition of eye tracking is especially important for accurately assessing when a user is looking at a real target, although it might be less important when they look at a virtual target [3].

Software

We developed a custom app for the Microsoft HoloLens that initiates a UDP socket and sends data about its location, rotation and all virtual objects to a predefined local IP address. A custom python server that runs on the computer with this IP address detects incoming data of all HoloLenses that send data and registers the devices. The continuous stream of data, reaching 60hz from each HoloLens, is forwarded from the server to all other connected devices at the same rate.

The server can be configured to send a calibration command to all HoloLens and pupil labs software simultaneously after which a calibration procedure starts. The HoloLens displays a calibration target at several locations sequentially. The pupil labs software receives the locations of the displayed target and correlates recorded pupil positions to calibration target positions with the assumption of the user looking at the target. Once there are 9 recorded positions the calibration procedure stops. Newly recorded pupil positions can be used to estimate the user's gaze relative to the HoloLens display. This gaze data is continuously queried by the server and send to the HoloLens that the eye-tracker is attached to.

The data-stream that a HoloLens sends and receives allows us to display virtual object manipulations to and from other users but also enables us to display the gaze points of other users in the same room. In our demo we displayed small spheres at the real-time estimated gaze point of other users using the head rotations from their HoloLens and pupil positions from the eye-tracker.

The server that processes all incoming and outgoing data can visualize the passing data. We can plot in real time a top-view or a 3d plot of virtual objects, participants and their gaze points locations.

Applications in Driving Research

As we argued in our work exploring the use of HoloLens to communicate with remote conversants while driving [2], drivers will likely appropriate AR technology for various uses in vehicles, including those for work and play [1]. We are currently planning to use the system introduced above to explore how such appropriations might affect a driver's ability to control the vehicle. We are particularly interested in how the use of AR technology might affect the ability of the driver to return to the driving task in partially automated vehicles.

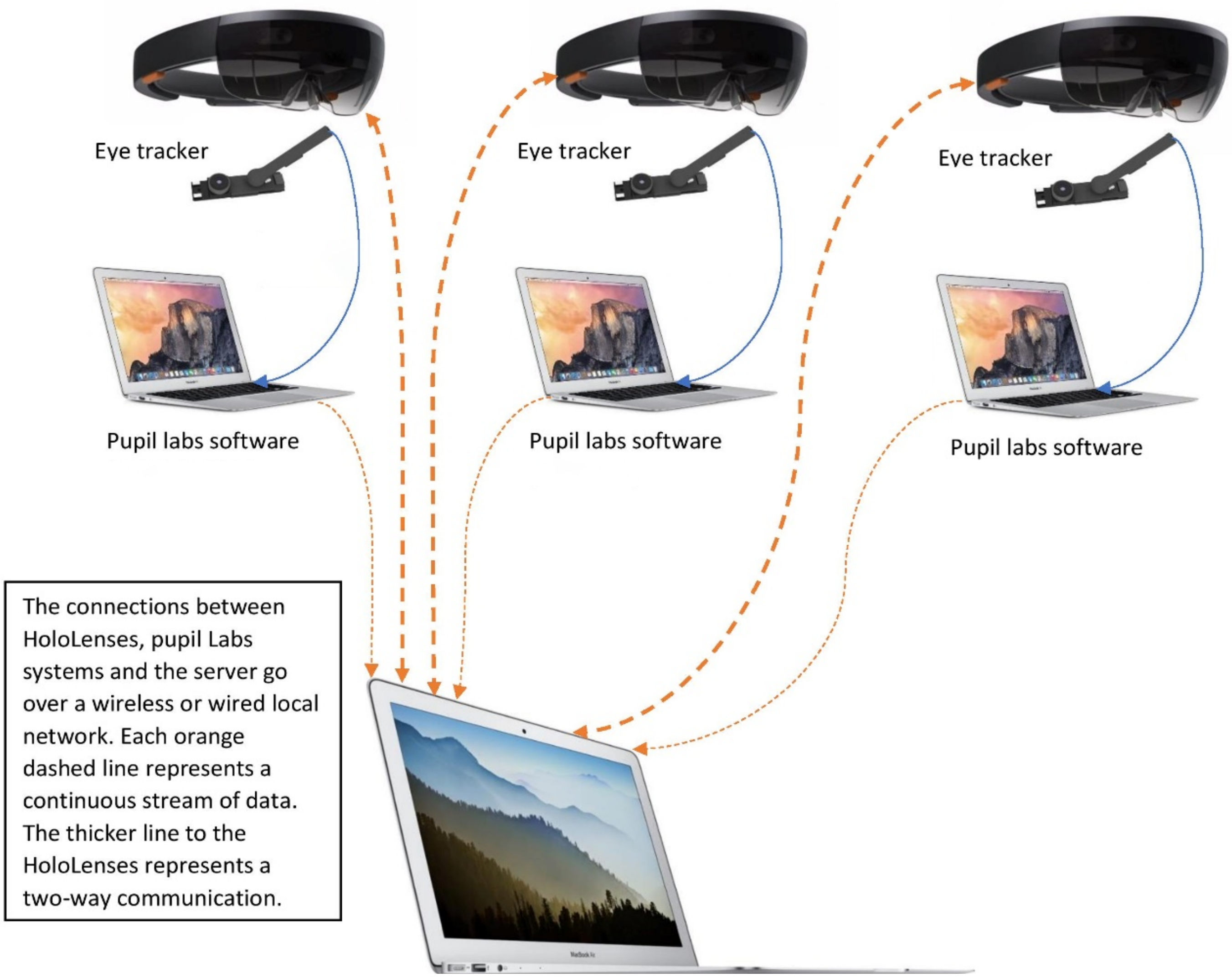

Figure 1. Overview of communication between the HoloLens devices, eye-tracker hardware and the server.

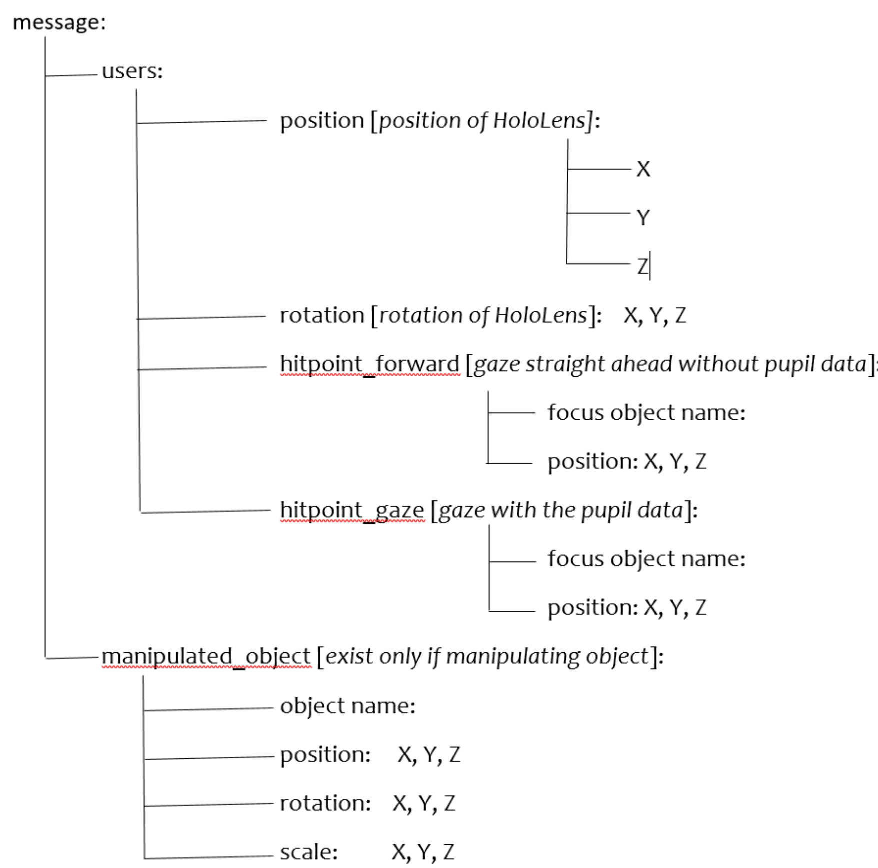

Figure 2. Overview of the data that a HoloLens sends to the server.

Limitations

During a small pilot study, we found that it is tricky to position the eye tracker at the ideal position for different participants. Sometimes the HoloLens forced us to position the eye tracker too low to get a good-enough image of the eyes, resulting in a low confidence. Future experiments could investigate ways to improve the confidence of the eye-tracking. We also found that the technology requires a stable WIFI network to maintain a smooth collaborative experience.

Acknowledgements

This work was supported in part by a grant from the UNH Broadband Center of Excellence.

References

[1] Andrew L. Kun, Susanne Boll and Albrecht Schmidt. 2016. Shifting Gears: User Interfaces in the Age of Autonomous Driving. *IEEE Pervasive Computing 15, 1 (2016), 32-38*.

[2] Andrew L. Kun, Hidde van der Meulen, and Christian P. Janssen. 2017. Calling While Driving: An Initial Experiment with HoloLens. *9th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design (Driving Assessment).*

[3] Hidde van der Meulen, Andrew L. Kun and Orit Shaer. 2017. What Are We Missing? Adding Eye-Tracking to the HoloLens to Improve Gaze Estimation Accuracy. *ACM International Conference on Interactive Surfaces and Spaces*.