Remapping on VR for hemianopia patients

Gonzalo Lorca

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

December 17, 2021

Abstract

abstract goes here

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: ____________________ Signature: ____________________

Acknowledgements

acknowledgements go here

Contents

Chapter 1

Introduction

Hemianopia is a visual impairment usually caused by stroke, head trauma or tumours wherein half of of the peripheral vision is lost in both eyes. It is one of the most common visual impairments worldwide, one Harvard study estimates that hemianopsia accounts for one third of vision rehabilitation patients in the US. [1] And considering that ten percent people develop hemianopsia after a stroke, it is clear that Hemianopsia affects and will continue to affect a lot of people worldwide. People with hemianopia report difficulty watching television, using mobile devices, reading, and navigating physical environments.[1][2] There have been many attempts to build tools to help hemianopia patients adapt. Some have proposed optical methods, for example X proposes the use of optical lenses that shift images in the centre of the visual field to the side of the eye. However, with the rise of head mounted displays over the last decade, there have been some efforts to create software based visual aids. A California State University study used Google glasses and machine learning software to find objects of interest in the patients blind area and project them where they can see them.[3] Solutions involving a projection of all the whole visual field are less common, this may be because AR devices like google glass are not well suited for projecting a whole visual field as VR HMD's would be, and these in turn tend to be less portable and tend to have smaller screen sizes themselves. It may also be that researchers have been discouraged by the negative side effects of optical methods that did something similar. Nevertheless, it is worthwhile to test solutions that project all of the contents of an expanded visual field. If successful, a solution like this one can help patients with tasks for which shifting the position specific objects into the users visual field may not be useful, such as reading, writing and interacting with PC's and other screens. Third paragraph. Proposal. A brief description of what we are going to build.

1.1 Proposal

To explore the utility of projecting an expanded field of view on a high definition head mounted display for aiding hemianopia patients with the identified difficulties, my plan is to produce a program that collects, transforms and projects live video footage onto a head mounted display. To make the program usable, additional components will be created to measure dimensions of the visual field of a patient and to measure the size of the lenses of any immersive HMD. The program will come with specifications for usability studies. A pilot study will be done with at least one test subject to test the potential for further research in this topic.

Chapter 2

Requirements

2.1 Functional Requirements

The program MUST, stream live images into the head mounted display (HMD) at a reasonable speed (not less than one second of latency)

The program MUST, transform frames before displaying them in the HMD.

The program MUST, have a perimetry reading functionality that locates the unusable area of the user's visual field.

The program SHOULD, have a setup protocol for new HMD's that requires little or no manual input.

The program CAN, have an initial menu window that allows users to select their profile with their details stored in it.

2.2 HCI Requirements

Using the program MUST not be exceedingly dizzying or disorienting.

The program SHOULD be able to improve the ability to read and write of users with hemianopsia, as well as other close up tasks.

2.3 Experiment Requirements

The experiment must show evidence for or against the usefulness of the remapping of images onto an HMD for people with hemianopia.

2.4 Hardware Requirements

2.4.1 Sensory Input Hardware

Any hardware used must be able to capture and live stream video footage from in front of the user. The aggregate fov of the cameras used should be large enough to capture the field of view that want's to be projected. In this case the objective is to recreate an ordinary field of view so the aggregate fov of the cameras should not be less than 180º horizontally.

2.4.2 Head Mounted Display (HMD)

The head mounted display must have at least two degrees of freedom. The HMD should be high definition, high enough for objects in projected images to be identifiable and usable. The HMD should have a wide usable field of view. A usable field of view is field of view of the HMD minus the sections of it that are demmed inacessible or unusable for any reason.

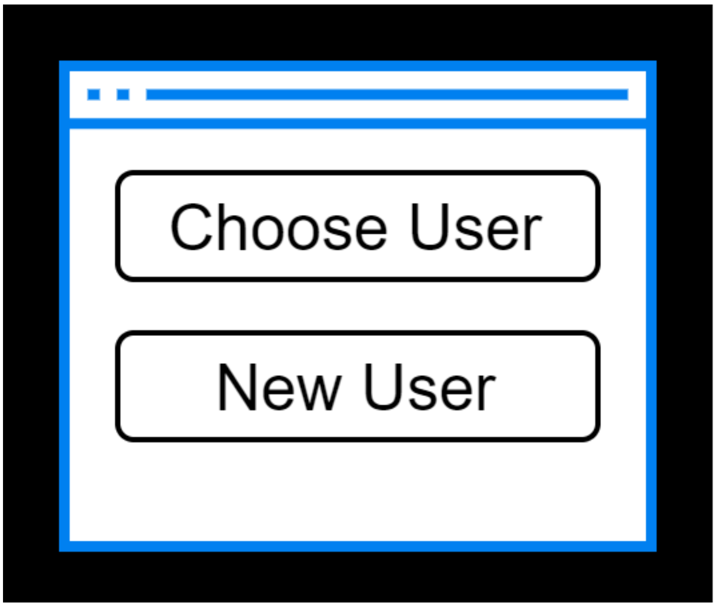
Chapter 3

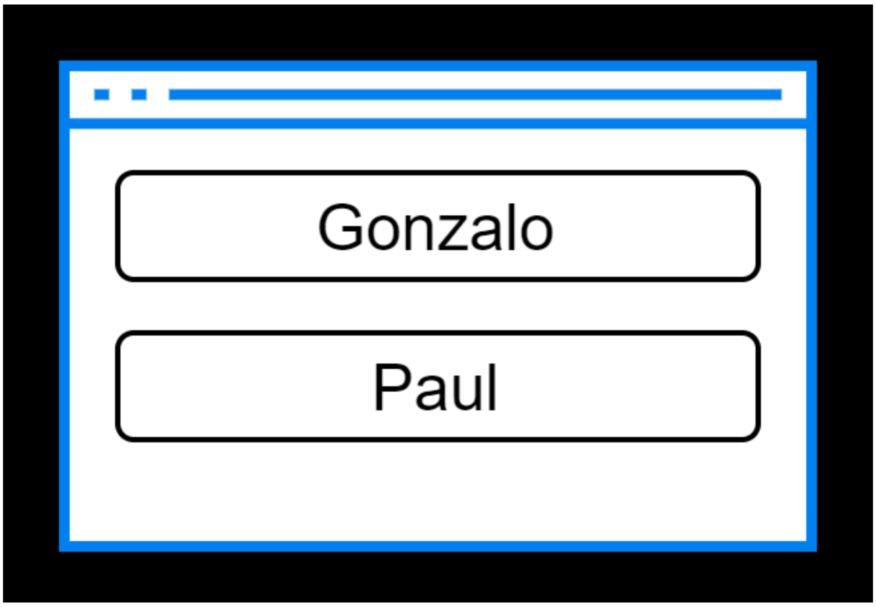
UI Design

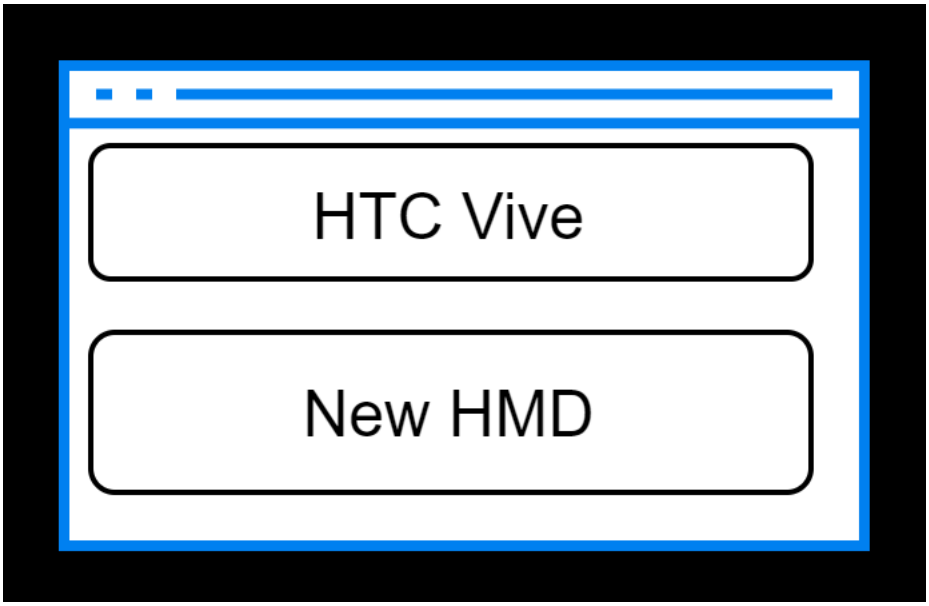
Making the program usable has two sections. One is the interactions the user has with the projections on the HMD, the other is the user interface around it that the user interacts with before putting it on and after taking it off.

3.1 Out of HMD interface.

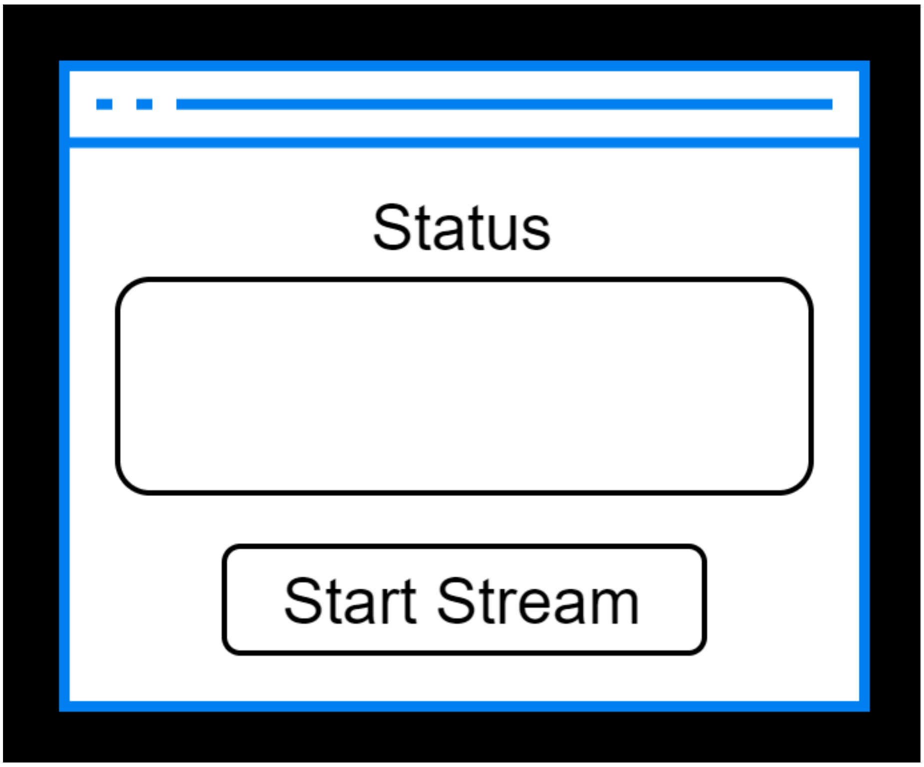
The program has a menu that allows the user to start the main program using the settings and the HMD that he has used before. The user can do this by clicking through to the his profile and then clicking on the Head mounted display they have used before.





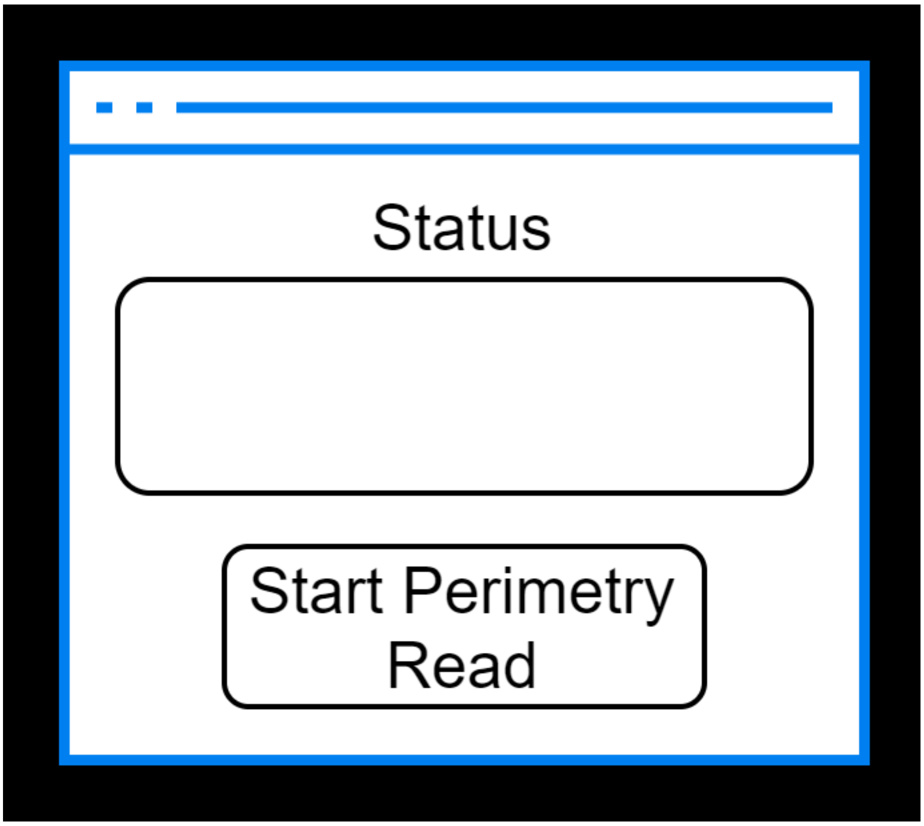


The user can choose a device that they have registered before. Clicking on it creates a new window for running the program.



Pressing start will run the program, and closing the window will stop it. Pressing start a second time will restart it. The status text box shows error messages to the user.

The user can also choose to register a new HMD which will take them to a form and then prompts the user to run a perimetry read on that device.This step is required for the program to know what areas of the HMD the user is blind to.



If the User is not registered, they can create a profile for themselves, which asks them to declare the type of hemianopsia they have. It then asks them to add a HMD to their profile and run the perimetry reading test on it.

3.2 HMD interface

There are two programs that the user will have the HDM on for. The perimetry reading script during setup and the live streaming program itself called 'Remapper.py'.

3.2.1 Perimetry Reader

The perimetry ready interface is a window that is as large as the entire HMD display. On it there is always a still red dot located at the center of one of the HMD's lenses and a moving black dot, which moves in the horizontal direction starting from the edge of the lense. Depending on the type of hemianopa of the user, the black dot may start on the right and move left or start on the left and move right. The user performs the operation one eye at a time, clicking a button to notify the program that the black dot is no longer in their field of view.

3.2.2 Remapper

The user interface for the main program is composed of two windows, on for each eye, that present live video footage to the user. These windows have no borders or aditional buttons, to close the interface the user must take off the HMD and close the start window on the out of HMD interface. The user adjust the position of the windows on the display manually if they want to, by using the arrow keys for the widow on the right eye and the ASDW keys for the left eye.

Chapter 4

Implementation

4.1 hardware

And HTC Vive was used for the displaying and two raspberry pi's were attached to it to collect the sensory data. A personal computer was used to collect the sensory information, process it and display it onto the HMD.

4.2 Raspberry Pi's:Collecting and Sending Frames

Two Raspberry pi computers are used. Each one operating a single camera and both of them running the same code. Two programs are used to collect and send data. One is the 'Camera Reader.py' script which collects frames from the camera using an open CV VideoCapture object, and puts them on a queue.The other is the 'Frame fetcher.py' which creates a Camera Reader object, starts it and runs it on a thread, connects to the PC via Socket connection, and then runs an infinite while loop that grabs a frame from the queue, serializes it, packages it and sends it to the PC.

4.3 PC:Transforming and projecting the frames

On the PC a single file called 'Remapper.py' is used to receive, transform and display the images to the user.

Chapter 5

usability testing

A full HCI experiment with test subjects was originally planed for this dissertation, the specification for it can be found in this paper's references. Unfortunately, unexpected complications in the development of the first prototype and time constraint of the dissertation has made the running of this experiment unfeasible. Instead, usability testing will be done with the

Chapter 6

Conclusion

Appendix A

First appendix

A.1 Section of first appendix

Appendix B

Second appendix

Bibliography