Adapting smart glasses for persons with loss of peripheral vision

HPR/D-006

by

Audun Borgersen, Jonas Claesson & Morten Grundetjern

Bachelor report for DAT 304, spring 2018

Advisor: Chief Engineer Jostein Nordengen

Faculty for Engineering and Science
Universitetet i Agder

Grimstad, may 2018

Status: Final

Keywords:
Smart glasses, peripheral vision loss, bachelor, Retinitis Pigmentosa, Vuzix M300, augmented reality.

Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	**Vi erklærer herved at vår besvarelse er vårt eget arbeid, og at vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.**	✔
2.	**Vi erklærer videre at denne besvarelsen:** - ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. - ikke refererer til andres arbeid uten at det er oppgitt. - ikke refererer til eget tidligere arbeid uten at det er oppgitt. - har alle referansene oppgitt i litteraturlisten. - ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	✔
3.	**Vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen** §§ 31.	✔
4.	**Vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.**	✔
5.	**Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.**	✔
6.	**Vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.**	✔

Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage Aura og på UiA sine nettsider med forfatter(ne)s godkjennelse.

Oppgaver som er unntatt offentlighet eller tausehetsbelagt/konfidensiell vil ikke bli publisert.

Jeg/vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering: ☑JA ☐NEI

Er oppgaven båndlagt (konfidensiell)? ☐JA ☑NEI
(Båndleggingsavtale må fylles ut)
- Hvis ja:

Kan oppgaven publiseres når båndleggingsperioden er over? ☑JA ☐NEI

Er oppgaven unntatt offentlighet? ☐JA ☑NEI
(inneholder taushetsbelagt informasjon. Jfr. Offl. §13/Fvl. §13)

Summary:

In our project we attempt to utilize advances in smart glasses and augmented reality technology to aid people with impaired vision. Specifically, people with retinitis pigmentosa and macular degeneration.

The problem is if/how can these patient groups use smart glasses to aid them in their everyday lives? What, if any, challenges do they have using the smart glasses?

Our solution is to develop an Android application for Vuzix M300 smart glasses that seeks to answer these questions.

We found that while our solution has flaws, both in software and in hardware, the technology itself has potential.

Preface

This bachelor project was assigned by Dr. Harald Reiso of Sørlandet hospital HF. This assignment related to the two diseases, retinitis pigmentosa and age related macular degeneration.The project assigner was interested in testing out whether or not AR glasses could be used to aid people with these types of vision impairments. They were originally interested in what smart glasses could do for people with AMD, but initial testing showed RP patients could perhaps better benefit from the glasses.

A similar project was undertaken recently by Kenneth Lorentzen titled “MANIPULERING AV VIRTUELLE BILDER FOR KOMPENSERING AV TAPT SYNSFELT” (“Manipulation of virtual images to compensate for lost field of view”) which aimed at something similar to this with VR goggles rather than AR glasses. While the results for the VR concept were somewhat negative, during testing for this project some minor trials were performed with project assigner in which some AR glasses were trialed, with promising results.

Based out of UiA Grimstad campus, we have worked to develop an app to the Vuzix M300 smart glasses that can help people with a loss of peripheral vision have an easier time navigating.

The program was written in Java using Android Studio along with the M300 SDK. it was then installed on a pair of M300, and then tested by two volunteers with differing visual impairments.

A thank to those who has helped us, such as Harald Reiso, Steinar Omnes, Chief Engineer Jostein Nordengen, Olav Nielsen, Britta Tranholm Hansen & Kenneth Lorentzen.

Grimstad

14. May 2018

Audun Borgersen, Jonas Claesson & Morten Grundetjern

Table of contents

Table of figures

1 Introduction

Humans throughout history have been reliant upon sight in order to function in society. As civilization has grown older, our dependence upon it has only increased. Be it to spot incoming traffic, read signs or to find environmental details like the button to open a buss door or even finding the door itself.

Luckily, as our reliance grew so has our ability to use tools in order to compensate for loss of natural eyesight. This project has aimed to add to these tools through an application for smart-glasses which allows persons with loss of peripheral vision such as those with Retinitis Pigmentosa a greater field of view. To a lesser extent the project has also looked into the use of an app to allow people with diseases such as Macular Degeneration to perceive details better, such as reading text.

1.1 Background

Augmented reality (AR) is a view of the physical world where the sight of reality is in some way changed or augmented. The first actual reference to something resembling a set of AR goggles came in 1901 within the book “The Master Key” by L. Frank Baum in which the protagonist is given glasses by a demon which allowed him to view a person’s character traits through letters displayed above their heads [16].

While not quite as demonic in nature as the glasses written about within The Master Key, humanity still eventually progressed to the point in which they could toy with the idea of creating similar technology. An early concept attempted to remotely control robots while still maintaining the operator’s sense of presence and coordination in 1993. [17] A diagram from this early experiment is shown below.

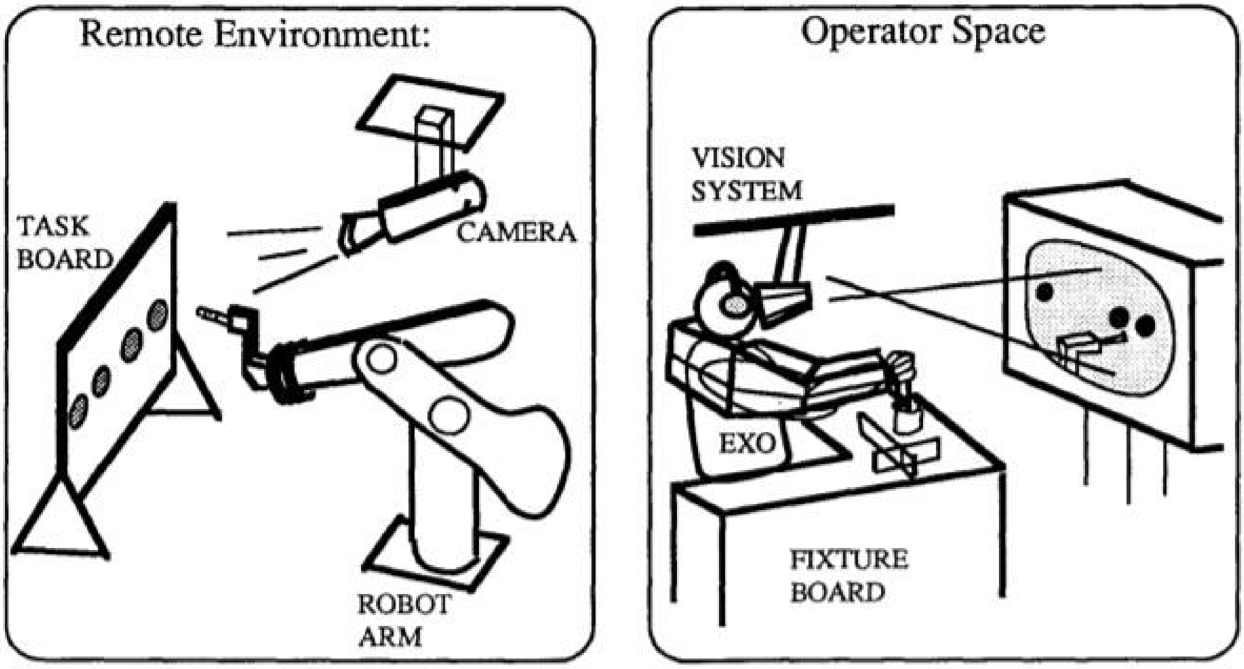

Figure 1.1:1 Operator testing Virtual Fixture [17].

These clunky prototypes eventually coalesced into the later sleek designs of products such as the Google Glass in 2013. The smart glasses as a concept had gained a roughly standardised shape which would be iterated upon over the next couple of years.[22] The Vuzix M100 was another version of this design, Its successor improving upon it with the release of M300 in the spring of 2017 [23].

This project builds upon the bachelor thesis of Kenneth Lorentzen, where he used VR goggles to compensate for sight loss in Virtual Reality. The HTC Vive headset was used to help several people with sight related diseases.

Our project aims to use smart glasses to aid the same patient groups. To investigate what glasses are best suited for this purpose, if and how they then can be of use.

There does exist some similar solutions publically available as of the writing of this report. Utilizing other smart-glasses or focusing on other aspects of the app. These will be looked into more in-depth within chapter 2.1 Similar Solutions On Market.

1.2 Problem definition

Those suffering from diseases such as those listed within chapter 2.3 Relevant Diseases usually struggle to navigate new environments as well as a sighted person would. They may find it difficult to read text from a bus schedule or spot incoming traffic.

This project aims to alleviate some of these disabilities through an app developed for the Vuzix M300 Ar glasses. The app does this by giving the user access to an additional screen with higher field of view and of greater detail than they would usually have.

In our preliminary report we mapped out a series of requirements and features for the application, while many of our estimations were off it is still a useful reference for planned features.

Within this context red page lists features we believe to be integral in getting our application to work. Green page contains features we reasonably expect to implement at some point in development. Black page means features which will consider implementing once we have time. Difficulty within this chart is refers to how difficult we believe implementing the specific features will be.

The numbers listed are arbitrary, only meaningful when considered relative to each other. Keep in mind that these numbers and features were first listed when creating this project’s preliminary report.

Figure 1.2:1 Development Schedule

Red page	Difficulty	**Green page**	Difficulty	**Black page**	Difficulty
Access Camera	3	Image manipulation	6	Lens attachment	3
Filters	6	OCR(Optical character recognition)	9	Boot to app	5
Brightness adjustment	3	Voice Recognition(Voice control)	6	Image warping for MD patients	10
Contrast	3	Talk back Gesture Interface	5		
Basic User Interface	6	Remote access	7		
Zoom	5	Pictures	3		
		Video	4		
		Tell Time	4		
		Display phone camera feed	3		

1.2.1 Functional requirements

Access camera: *The first feature we are to implement where we can display the camera feed from the smart glasses to the screen.*

Basic user interface: *A way to access the different features*

Filters: *Filters that display the video feed from the camera in different colors. May help the person with the glasses to see details better.*

Brightness adjustment: *The ability to change the brightness on the screen*

Contrast: *Increase or decrease the level of contrast on the screen*

Zoom: *A digital zoom where the screen zoom in on what the user want to see.*

Image manipulation: *An ability to change the image on the display so that we could tailor the screen to the person's vision*

OCR (Object Character Recognition): *The smart glasses camera would take a picture and recognise the characters in the picture then the glasses would read the text for the user.*

Voice recognition: *The glasses would recognize certain command words and it would allow easy access to features*

Remote access: *Gives the possibility to see through the phone camera.*

Talk back gesture interface: *Android devices have built in accessibility functionalities, implementing these on the remote control can prove advantageous.*

Pictures: *An ability to take a picture with the glasses.*

Video: *An ability to capture video with the glasses.*

Tell time: *A feature that would read the current time for the user*

Display phone camera feed: *Display a smart phone’s video feed on glasses’ screen.*

Lens adjustment: *Look into external lens attachment to further expand the field of view.*

Boot to app: *The application would start immediately upon glasses boot to let the user avoid navigating the Vuzix base menu for each use.*

Image warping for MD patients: *A symptom of wet AMD is that ones vision becomes "twisted". Straight lines appear to be bent. Could the camera feed be manipulated to compensate for this?*

1.2.2 Additional functional requirements created during project

Color Temperature: *Adjust the screen's color*

Barcode scanner: *The ability to scan a product's barcode.*

Save settings: *Save relevant settings between sessions.*

Text to speech: *tells the user relevant information*

Tell the Date: *gives the user information about the date*

Flashlight: *allow the user to turn the glasses' flash function on in order to provide a constant source of light.*

1.2.3 non-functional requirements

Performance: *The app relies upon the user perceiving the world through the screen of the glasses, we therefore believe that a noticeably poor framerate will be detrimental to the user's experience.*

Usability: *As the primary demographic of the app may may have some difficulty reading more traditional user interfaces, we believe that investing time into creating accessible ways to interact with the will be worthwhile.*

Stability: *A priority with most apps is to prevent unnecessary crashes, even more so when users may be using it to navigate unfamiliar environments.*

1.3 Prerequisites & limits

1.3.1 Limits in hardware

Our choice of hardware was limited to the hardware that was provided by the hospital which were the eSight 3, Vuzix M100 and Vuzix M300 glasses.

Several key factors determined the choice of hardware:

- User feedback: Prior to our bachelor the hospital have had a test where they had tried out the different hardwares the hospital had brought in. They said that the esight 3 was too big and unwieldy. Some of the testers with RP were positive to the M100 glasses.

- Openness: M100 and M300 runs on Android version 4 and 6 respectively. Whereas eSight 3 is its own proprietary, closed system. Vuzix also offers an official SDK.This allowed us to develop an application for the glasses similar to how you can develop an app for mobile phones.

- Technical specifications: Comparing the datasheets for the M100 and M300 [12] [13]. The M300 appears to be an upgrade from the M100, particularly in terms of FOV (20 degrees vs. 15 degrees). This is especially important, since the aim of the project is to expand someone's FOV. It is noteworthy that eSight 3 has a FOV of 37 degrees, but is a non-starter for us after revealing itself to be a closed platform.

In the end, the Vuzix M300 smart glasses became our AR glasses of choice. Its software being open and modifiable as well as being technologically superior in terms of specifications to the m100 glasses.

1.3.2 Limits in software

Due to time constraint and convenience, we will have to resort to utilizing third party open source software for certain features. For instance we will not make our own optical character recognition engine, but instead use open source alternatives such as Tesseract.

Also the software the Vuzix M300 glasses are running on isn't completely compatible to the software that runs on a standard phone. Because it lacks services like the google play services which means we can not use the inbuilt functionality of the android libraries, such as androids speech recognition.

In addition, Vuzix speech recognition is limited by the trigger word it uses. A user has to say the phrase “hello vuzix” each time before making a new command. This key phrase allows for a short time frame in which several commands can be spoken.

1.3.3 Other limits

The Eye condition we are targeting is not common, we have had some difficulty finding other volunteers with the relevant diseases. To compensate for this lack of examinees we will focus on a smaller group of people to test it out, but we will spend more time on each one.

1.3.4 Prerequisites

This project ultimately rests upon the assumption that it will, in some way, be helpful to people with the aforementioned eye conditions to use the camera feed of a pair of smart glasses to expand their field of view or to perceive detail.

1.4 Literature study

A useful reference has been the bachelor report of this project’s precursor report by Kenneth Lorentzen which dealt with a similar problem through the development of a VR (virtual reality) application for the HTC Vive VR headset.

Upon the start of the project we were sent a fair amount of introductory information through email by Harald Reiso regarding the targeted diseases and the work done so far.

Most of the code was created using either Android’s proprietary documentation within Android Developers [18] or Vuzix Developer centre [19] (login required]). Further we utilized some code examples from sites like stack exchange and similar. Links to these example can be found below this report’s references.

While considering the use of different smart-glasses early in the project a Youtube channel called “The Blind Life” was a useful reference for an unbiased blind person’s account of the usability and usefulness of several competing models of medicinally minded AR devices.[20]

1.5 Problem solution

We wish to help increase the navigational skills of persons suffering from RP and AMD.
Our solution to this problem is by utilizing the unique screen placement of the Vuzix M300 glasses. Since the M300 glasses then allows the user to use their natural eyesight most of the time and only

glance up into the screen once necessary, this made wearing the glasses comfortable in realistic settings. This also allows for the user to have direct eye contact with other people when engaged in a conversation.

Our primary goal is to alter the image so that the user can get the most information about the environment from the glasses. This was planned to be achieved through brightness, contrast and color filter settings, as well as being able to adjust what portion of the screen the camera preview was displayed on. Additional potentially useful features were planned in order to add some more concrete functionality to the app. These included features like zoom and OCR.

This application will be created through the development of an Android app designed for the mentioned Vuzix M300 glasses. The specifications as to what features are planned for this app is listed within subchapter 1.2 Problem Definition.

Testing the viability and usefulness of certain features and the concept as a whole will be done through careful demo sessions with two examinees who have volunteered to help us. One of these examinees suffer from Retinitis Pigmentosa, the other from Retinal Dystrophy.

1.6 Report structure

In the next chapter we will be discuss existing products and theoretical background, like other smart glasses and the previous bachelor project this thesis is based on.

Chapter 3 details the specifics of our solution, its features and capabilities. In it, we will discuss how we aim to address certain problems and worries that appeared in the previous chapters.

We will discuss and reason over our discoveries during testing in chapter 4. There is little value in this research and development without testing. The result of how the features showcased in the preceding chapter worked in the field is elaborated on here.

Then we will conclude our report in chapter 5, where we summarize our work and findings in the two foregoing chapters.

2 Theoretical background

The concept of using technology to enhance vision, or to recuperate lost vision, is not a new one. From depictions in fiction, like Geordi La Forge's visor,

Figure 2:1 LeVar Burton as Geordi La Forge [11]

to present day, real products like eSight 3 [4]. Using some form of technologically advanced glasses to aid or enhance sight seems like an obvious idea that has not been feasible until recently.

Figure 2:2 Man using eSight 3 [4]

These sort of glasses are nowadays referred to as either smart glasses or AR glasses, a smart glasses is a set of glasses with an integrated computer and screen which may overlay information of a camera feed through a HUD or similar. In this context AR glasses stands for Augmented Reality glasses and is interchangeable with smart glasses.

2.1 Similar solutions on the market

There does exist several competitors within this field, though we found none who targets the exact niche that is this project’s main demographic: people with RP.

eSight 3 [4]

Figure 2.1:1 eSight 3 [4]

An eye covering visor with a claim to aid with an assorted number of vision impairments [5]. eSight 3 glasses offer features such as bioptic tilt that allows for the user to tilt the glasses while in use to switch between the visor and the user's natural sight. Its price point sits well above other solutions presented, at a hefty 9995$ [21].

IrisVision [6]

Figure 2.1:2 Woman trying out IrisVision [6]

Combining a Gear VR headset [10] and a Samsung Galaxy S7, Iris Vision's device has a high FOV.

The headsets utilise the power of the phone's camera and have a patented bubble zoom feature that allows for parts of the screen to be zoomed in while the rest of the screen stay at default level of zoom, so the user can maintain their situational awareness while zoomed in. Retails for 2500$ [6].

MyEye 2.0 [7]

Figure 2.1:3 Hands holding a MyEye 2.0 [7]

This device is technically not a pair of smart glasses, but is a camera designed by Israeli wearable technology company OrCam. The camera can assist people with vision impairments by the use of OCR, and can recognise faces. It relays the information through a speaker close to the wearer's ears [7].

Nueyes pro [9]

Figure 2.1:4 NuEyes Pro

These AR glasses are a pair of R-7 smart glasses made by Osterhout Design Group [15].

2.2 Preceding project

Project assigner had requested an earlier bachelor project regarding this same subject. Kenneth Lorentzen's "MANIPULERING AV VIRTUELLE BILDER FOR KOMPENSERING AV TAPT SYNSFELT".

The primary difference between Lorentzen's project and ours being the fact that instead of AR it revolved around creating a virtual reality app with a similar purpose to our project.

Ultimately the project concluded that while VR could be useful for some patients in order to get a better view of an image, none of them were able to read text in virtual reality. Further, the report discovered that yellow or white text on a black background was the most comfortable for patients to view. AMD patients found contrast to be the most useful feature displayed. Finally that a set of VR goggles could be used by RP patients as a replacement for computer monitors as the goggles allow for light to be controlled thoroughly.

2.3 Relevant diseases

2.3.1 Retinitis Pigmentosa (RP)

Retinitis Pigmentosa is a collection of genetic eye conditions where the cells in the light sensitive retina slowly dies.

The symptoms commonly associated with RP is reduced dark vision, light sensitivity, and slowly the loss of peripheral vision until the patient is left with only a “tunnel vision”.

Patient with this condition find it difficult to navigate their surroundings, as they have to scan the room with their central vision to get the full picture. Approximately 1 in 4000 people develop RP [1].

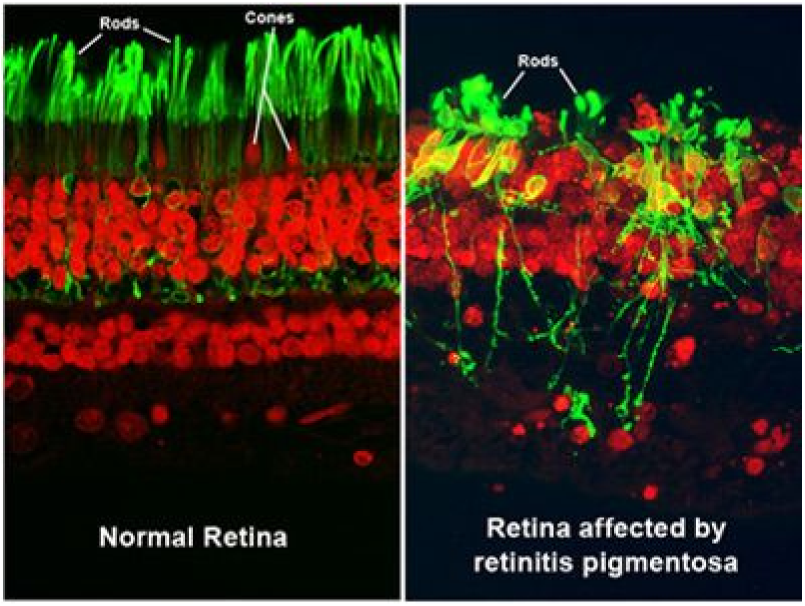

Figure 2.3:1 Image of a normal retina compared to one affected by RP[1].

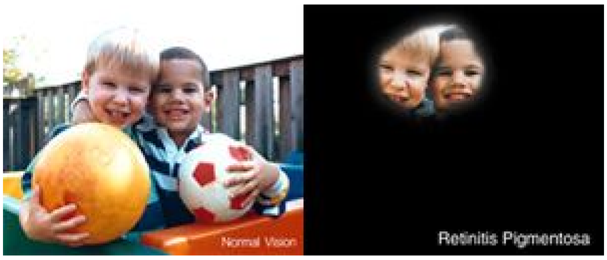

Figure 2.3:2 Picture illustrating loss of field of view [1].

2.3.2 Macular Degeneration (MD)

A common cause of loss of vision among people aged 50+. Macular degeneration is a medical condition which damages the macula, the cluster of cells on the retina responsible for central vision, eventually resulting in a blurry spot in the center of vision.
While it will not trigger complete blindness, it might make daily tasks such as recognizing faces, reading, driving or similar tasks difficult.

The exact cause of AMD is a large buildup of drusen, buildup of deposits of a fatty protein called lipids. Two subgroups of MD exists, Geographic atrophy (dry AMD) and neovascular (wet) AMD. Dry AMD is when the cells in and surrounding the macula breaks down, causing loss of vision. Wet AMD on the other hand, is when new blood vessel grow in an abnormal way. These vessels can then leak and burst, damaging the macula [2] [3].

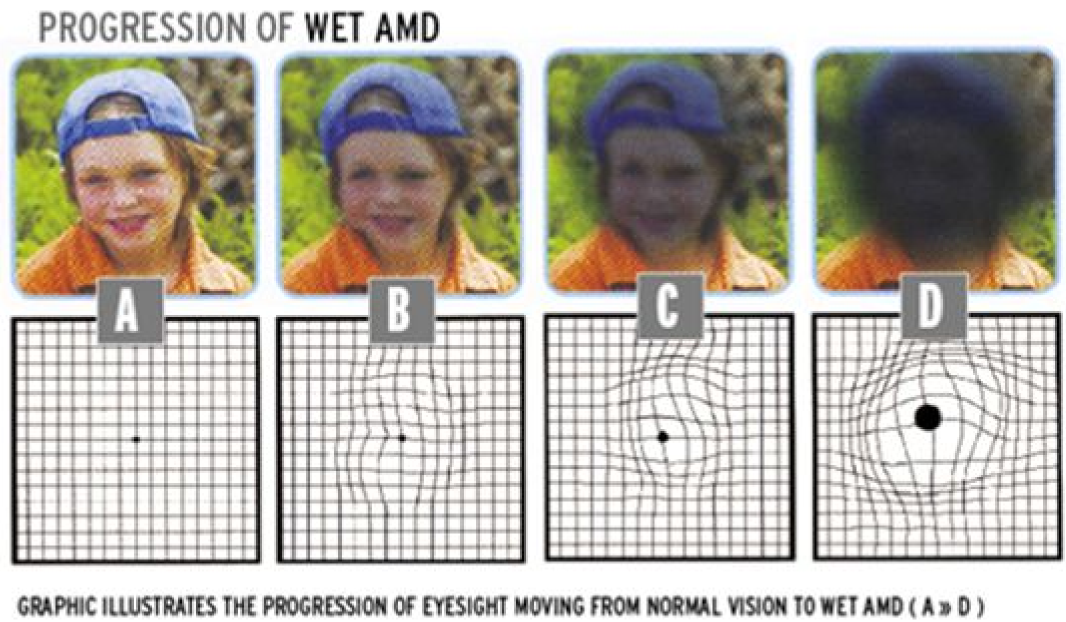

Figure 2.3:3 Progression of wet AMD [2].

3 Solution

This chapter will serve as a more in-depth explanation for our Android application. What it can help with, what features it provides and how it does both.

3.1 Requirement

The results for the implementation of each of the individual features.

3.1.1 Red page features

Access camera: *The first feature we are to implement where we can display the camera feed from the smart glasses to the screen.*

This was the feature we implemented first, as this serves as the project’s core.

Basic user interface: *A way to access the different features*

This is done through the use of a state machine. Each state corresponds to a mode that can then be altered as needed.

Filters: *Filters that display the video feed from the camera in different colors. May help the person with the glasses to see details better.*

The filter was implemented quite early in the process as it was a part of camera2 API. The M300 glasses supported 4 different color filter modes.

Brightness adjustment: *The ability to change the brightness on the screen*

The second feature to be implemented. It was implemented through the use of camera2 API.

Contrast: *Increase or decrease the level of contrast on the screen*

We found during development that it appears that the M300 does not support altering contrast.

Zoom: *A digital zoom where the screen zoom in on what the user want to see.*

The M300 camera lacks optics for zoom. So a digital one, via the camera2 API, was deemed handy and was implemented early.

3.1.2 Green page features

Image manipulation: *An ability to change the image on the display so that we could tailor the screen to the person's vision*

Abandoned due to poor feedback.the screen on the m300 is too small for image manipulation to be effective.

OCR (Object Character Recognition): *The smart glasses camera would take a picture and recognise the characters in the picture then the glasses would read the text for the user.*

We had it implemented for a while but had to abandoned it due to instability.

Voice recognition: *The glasses would recognize certain command word and it would allow easy access to features*

Using Vuzix's own speech recognizer, we established keyphrases for jumping between states, toggling the flashlight and telling time and date.

Remote access: *Gives the possibility to see through the phone camera.*

Abandoned due to end-user complexity.

Talk back gesture interface: *Android devices have built in accessibility functionalities, implementing these on the remote control can prove advantageous.*

Mostly abandoned due to lack of remote access. Something similar was done by simply having the device read relevant information aloud when relevant.

Pictures: *An ability to take a picture with the glasses.*

Was implemented at one point within development, but removed due to the unnecessary increase in complexity for end user.

Video: *An ability to capture video with the glasses.*

Abandoned for similar reasons as pictures.

Tell time: *A feature that would read the current time for the user*

A simple voice command will read the time or date aloud for the user.

Display phone camera feed: *Display a smart phone’s video feed on glasses’ screen.*

This feature could make it easier to read the phone text but there already exist several good assistant tools for this so we decided to scrap it.

3.1.3 Black page features

Lens adjustment: *Look into external lens attachment to further expand the field of view.*

We have chosen not to spend any time to experiment with hardware modifications.

Boot to app: *The application would boot immediately when the glasses was started so that the user would not need to go through the menus to open the app.*

Increased priority and implemented due to testers’ wishes.

Image warping for MD patients: *A symptom of wet AMD is that ones vision becomes “twisted”. Straight lines appear to be bent. Could the camera feed be manipulated to compensate for this?*

A lot of work to implement and low priority meant we never got to this. Could possibly have been a bachelor in it self.

3.1.4 Extra features implemented while working on the project:

Color Temperature: *Adjust the screen's color*

Feature implemented, but removed due to lack of usefulness

Barcode scanner: *The ability to scan a product's barcode.*

A feature we wished to implement due to input from testers. Vuzix developer centre had a example project built around this feature listed in the Vuzix developer centre, but we experienced some issues when attempting to implementing this feature and therefore abandoned it due to time constraints.

Save settings: *Save relevant settings between sessions.*

A simple quality of life setting implemented a short time into development when its usefulness became obvious. Used to store brightness, color filter and volume between sessions.

Text to speech: *tells the user relevant information*

Integrated within user interface in order to make it more user friendly.

Tell the Date: *gives the user information about the date*

Came as a by product of tell time feature.

Flashlight: *allow the user to turn the glasses' flash function on in order to provide a constant source of light.*

Implemented in relation to testing in dim or dark environments.

3.1.5 Non functional requirements:

Performance: *The app relies upon the user perceiving the world through the screen of the glasses, we therefore believe that a noticeably poor framerate will be detrimental to the user's experience.*

Performance was mainly achieved through the use of camera2 functions rather than the original camera functionality. In addition to avoiding too resource intensive features such as those which would require adjusting the values for each individual "pixel" per frame.

Usability: *As the primary demographic of the app may may have some difficulty reading more traditional user interfaces, we believe that investing time into creating accessible ways to interact with the will be worthwhile.*

Usability was achieved through using as little text throughout the app as possible, replacing it with the glasses speaking relevant phrases aloud. In addition the app was caused to start immediately upon boot in order to prevent the user from having to interact with the glasses' interface as much as possible.

Stability: *A priority with most apps is to prevent unnecessary crashes, even more so when users may be using it to navigate unfamiliar environments.*

Through internal testing we discovered a couple of system breaking bugs, but solved them without much issue. Our implementation of OCR was somewhat unstable and therefore removed.

3.2 Design specification

Developing for the Vuzix M300 glasses is more or less functionally identical to developing for Android, we have therefore chosen to work with the Android Studio IDE as this was designed for Android development and is the primary platform for such projects. Similarly, Java was chosen as our programming language as this was the most popular choice on the platform and each of the developers had a fair amount of experience with it.

The Vuzix M300 glasses has three buttons on its side that we use. It also has a touchpad and a microphone we use as input for our app. The buttons have two types of inputs: press and hold, and click.

The simple menu system allows for more modifiability because it is easy to implement a new state but there is a cost to every new state in the form of usability where every new state we add becomes a new state to go through when using the app.

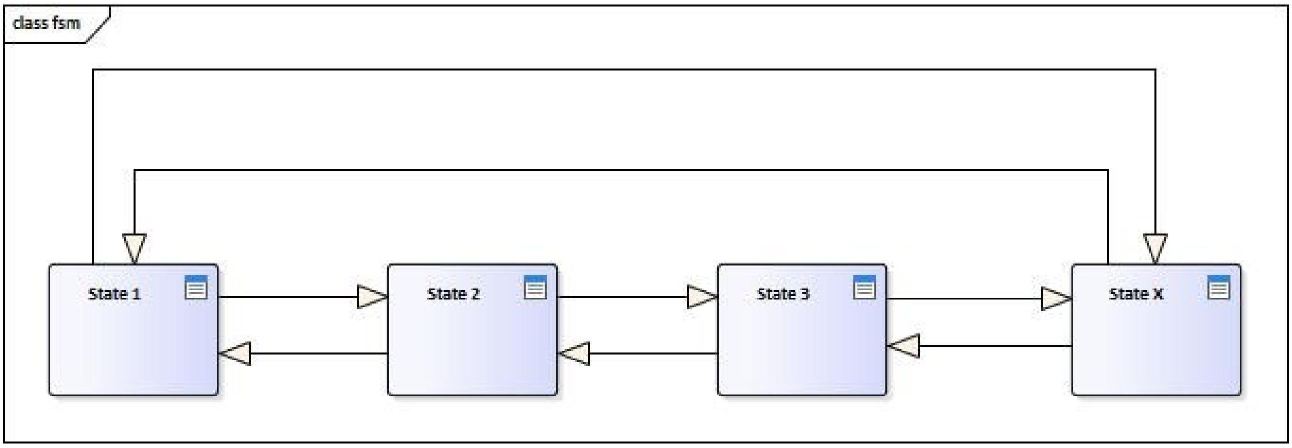

Figure 3.2:1 State machine for an arbitrary amount of states

The way we try to counter this is by clumping the most often used states together so that when using the app you would only go through the whole list when you want to change some setting.

The performance requirement was that when the user was using the device the user would not experience lag, utilizing Android camera2 was a major boon to performance.

3.3 Implementation

A source number is listed for some of the project's features. This means that we have used a coding example for this particular feature outside of the documentation provided by Android or Vuzix. A link to this example can be found within chapter 6 References.

3.3.1 The state machine

The menu system in the app are made up of states in which the different states allows access to and adjustment of different features.

This was done in order to keep the app compatible with the lack of user input methods, disregarding voice controls the user has essentially 6 possible ways to interact with the system.

```java
//Constructor
public StateInfo(float curr, float def, float min, float max,float inc, String name){
    this.currentVal = curr; //Value the state is adjusted by
    this.defaultVal = def; //Value returned to upon reset
    this.minVal = min; //lowest possible value for state
    this.maxVal = max; //highest possible value the state can be adjusted by
    this.incVal = inc; //the amount current value is increased by per increment
    this.stateName = name; //name of the state, this value is also used as save key and spoken aloud upon state switch
}
//adjusts state setting by increasing current value by set incrementing value, exceeding max values sets it to minimum value.
public void incVal(){
    currentVal = currentVal + incVal;
    if(currentVal>maxVal){currentVal = minVal;}
}
//adjusts state setting by decreasing current value by set incrementing value, values below minimum value set it to max value.
public void decVal(){
    currentVal = currentVal - incVal;
    if(currentVal<minVal){currentVal = maxVal;}
}
//Overloads reset without any parameters in order to make use slightly cleaner
public void reset(){this.reset( flag: false);}
//sets current value to defined default value, if default value already equals default value, current value is set to saved value
public void reset(boolean flag){
    //removing flagg val may cause some strange behaviour for users upon multiple resets, unadjusted settings jumping to previously save points
    if(currentVal == defaultVal && flag == false){
        currentVal = savedVal;
    }
    else {
        this.savedVal = this.currentVal;
        this.currentVal = defaultVal;
    }
}
```

Figure 3.3:1 Code StateInfo class

One of these options is the menu button, something we have been unwilling to overload due to it being the only remaining method the user has to access the m300 glasses' system menu. This left five inputs, two were used to iterate between the different states in the app, two were used to increment and decrement the values within the state and the final input is used as a universal reset to default value for either each individual state value or every single state.

The state system is built around a class called StateInfo, upon bootup the program creates an object of StateInfo for each state in the app and sets the relevant values, either save system or constants. These objects are then placed within an array which is iterated through via the global currentState value.

3.3.2 The camera

The app camera is built around the Android class camera2, which is a significant improvement upon the original camera in terms of performance. Upon startup of activity a Camera2 CaptureRequestBuilder is created which records adjusted

```java
try {
    cameraCaptureSessions.setRepeatingRequest(captureRequestBuilder.build(), listener: null, handler: null);
} catch (CameraAccessException e) {
    e.printStackTrace();
}
```

Figure 3.3:2 Settings implemented programmatically

settings for camera preview before displaying changes. Brightness is then adjusted via setting the captureRequestBuilder’s `CaptureRequest.CONTROL_AE_EXPOSURE_COMPENSATION` setting with the variables in the brightness stateInfo object. Similarly the camera’s supported color filters are set with `CaptureRequest.CONTROL_EFFECT_MODE` through the currentVal variable in the colorFilter stateInfo object. Through testing we’ve discovered that the supported color effects for the M300 camera are MONO, NEGATIVE and SEPIA.
The flashlight feature is set through such settings as well, using `CaptureRequest.FLASH_MODE` to set it to `CameraMetadata.FLASH_MODE_TORCH.`

3.3.3 Boot to app

The app will open immediately upon startup in order to minimize unnecessary interface required to navigate for the user. This was done through a receiver in the Android manifest file.

```
<receiver
    android:name=".StartupReceiver"
    android:enabled="true"
    android:permission="android.permission.RECEIVE_BOOT_COMPLETED" >
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</receiver>
```

Figure 3.3:3 Android Manifest Receiver

The receiver listens for a BOOT_COMPLETED broadcast action which
is called once upon startup. Opening the StartupReciever class within the app which then initiates the app’s main CameraMain activity. This StartupReciever class extends BroadcatReciever in order to be capable of doing this.

3.3.4 Storage system

Early in the project we realized that the ability to store values between sessions would be quite valuable.[27]

The storage class works by opening a stream to the internal storage . The internal storage allows you to store data directly on the device and its only accessible from within the app.
The storage class has 2 functions save() and load.

```java
// Save the value at the key
public void Save(String Key,String Value, Context context){


    String fileContents = Value;
    FileOutputStream outputStream;
    try {
        outputStream = context.openFileOutput(Key, Context.MODE_PRIVATE);
        outputStream.write(fileContents.getBytes());
        outputStream.close();
    } catch(Exception e)
    {
        e.printStackTrace();
    }

}
```

Figure 3.3:4 Save function

save allows you to save a string by calling storage.Save(“key”,”value”) where value is the string you save and key is a string you use to get the value from the load function.

```java
// returns the string  stored at in the file
public String Load(String Loadfile,Context context){

    String returnVal ;

    try {
        StringBuffer stringBuffer = new StringBuffer();

        FileInputStream Input =  context.openFileInput(Loadfile);
        InputStreamReader Reader = new InputStreamReader(Input);
        BufferedReader Buffer = new BufferedReader(Reader);
        String Message = "";


        while((Message = Buffer.readLine())!= null) {
            stringBuffer.append(Message + "\n");
        }
        returnVal = stringBuffer.toString();


        return(returnVal);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
        return("File not found");
    } catch (IOException e) {
          e.printStackTrace();
        return("somting went wrong white loading");
    }
```

Figure 3.3:5 Load function

you use the load function by inserting the key you used to save the value in to the storage.load(“key”) and it return the the value as a string.

This class is used to store the different values in the states such as Brightness, color filter and volume.

3.3.5 Zoom

The zoom was one of the first green page task we accomplished. For the camera to zoom it was required to have an instance of rectangle to represent the area of the screen the digital should zoom in to.

We made a zoom class where we made 2 functions.

One to instantiate the zoom object

```
public zoomObject(float zoomMax, int maxX, int maxY) {

    this.maxX = maxX;
    this.maxY = maxY;
    this.zoomMax = zoomMax;
    this.zoom = 1;
    rect = new Rect( left: 0,  top: 0, maxX, maxY);
    centerX = rect.centerX();
    centerY = rect.centerY();

}
```

Figure 3.3:6 zoomObject

The second function changed the size of the rectangle to represent the zoomed in state the screen should display, and returned the new rectangle.

```
public Rect zoom(int zoomFactor) {
    if ((zoom * zoomFactor < zoomMax) && (zoom * zoomFactor > 1)) {
        zoom = zoom * zoomFactor;
        int WidthHalf = (maxX / zoom / 2);
        int HeightHalf = (maxY / zoom / 2);
        int tempCenterX = centerX;
        int tempCenterY = centerY;
        rect.set(
                left: tempCenterX - WidthHalf,
                top: tempCenterY - HeightHalf,
                right: tempCenterX + WidthHalf,
                bottom: tempCenterY + HeightHalf);
        centerX = rect.centerX();
        centerY = rect.centerY();
        return rect;
    }
    return rect;
}
```

Figure 3.3:7 zoom

The camera2 api has a function

```
captureRequestBuilder.set(captureRequest.SCALER_CROP_REGION,
        zoom.zoom((int) stateArr[currentState].currentVal));
```

that the second parameter takes a rectangle which our zoom.zoom(int zoo,factor) returns

The zoom function allows the user to digitally zoom inn on the picture up to 16 times. this can help the user to read text or to view objects far away.[25]

3.3.6 Voice commands

The voice command allows the user to usher commands to the glasses through the use of the inbuilt microphone. This make it possible for the user to use the application handsfree. The Voice commands were implemented through the integrated vuzix speech recognition.
The user need to say the trigger word “hello vuzix” then the device will listen for command words for a set amount of time.

```
sc.insertKeycodePhrase( s: "zoom", KeyEvent.KEYCODE_H);
sc.insertKeycodePhrase( s: "zoom out", KeyEvent.KEYCODE_I);
sc.insertKeycodePhrase( s: "test", KeyEvent.KEYCODE_G);
sc.insertKeycodePhrase( s: "time", KeyEvent.KEYCODE_A);
sc.insertKeycodePhrase( s: "dato", KeyEvent.KEYCODE_B);
sc.insertKeycodePhrase( s: "light on", KeyEvent.KEYCODE_C);
sc.insertKeycodePhrase( s: "light off", KeyEvent.KEYCODE_D);
sc.insertKeycodePhrase( s: "light up", KeyEvent.KEYCODE_E);
sc.insertKeycodePhrase( s: "light down", KeyEvent.KEYCODE_F);
```

Figure 3.3:8 Keycode phrases

If the glasses hear one of the command word it will trigger a key event we have specified to the command word.

```
if(keyCode == KeyEvent.KEYCODE_A) {
    Toast.makeText( context: this, Calendar.getInstance().getTime().toString(), Toast.LENGTH_SHORT).show();
    //Toast.makeText(this, df.format(Calendar.getInstance().getTime()), Toast.LENGTH_SHORT).show();
    tts.speak( text: " the clock is "+dfh.format(Calendar.getInstance().getTime())
            +"hours and  "+dfm.format(Calendar.getInstance().getTime())+ "  minnuts" , TextToSpeech.QUEUE_ADD,  params: null);
}
if(keyCode == KeyEvent.KEYCODE_B) {
    Toast.makeText( context: this, Calendar.getInstance().getTime().toString(), Toast.LENGTH_SHORT).show();
    //Toast.makeText(this, df.format(Calendar.getInstance().getTime()), Toast.LENGTH_SHORT).show();
    tts.speak( text: "it is "+dfd.format(Calendar.getInstance().getTime()) , TextToSpeech.QUEUE_ADD,  params: null);
}
if(keyCode == KeyEvent.KEYCODE_C) {


        captureRequestBuilder.set(CaptureRequest.CONTROL_AE_MODE, CameraMetadata.CONTROL_AE_MODE_ON);
        captureRequestBuilder.set(CaptureRequest.FLASH_MODE,CameraMetadata.FLASH_MODE_TORCH);


}
if(keyCode == KeyEvent.KEYCODE_D) {

    captureRequestBuilder.set(CaptureRequest.FLASH_MODE, CameraMetadata.FLASH_MODE_OFF);
```

Figure 3.3:9 Key events

We made this function originally by looking at the documentation on the vuzix sites. but an update broke our original code we manage to fix it by using the vuzix speech recognition example found on vuzix developer center. This code is listed as open source, “as is”. This means that use of the code by third parties is allowed as long as the requirements set by a disclaimer listed within the code is followed.

3.3.7 Volume

To make it easier for user with ocular disability we implemented a text to speech methods that would tell the user what state they are in and what value some settings are set to.
The default of text to speech is set max volume so we implemented a state that can adjust the volume to the users needs.

The voice control worked in a similar way where the key codes are triggered when the glasses hears a command phrase.

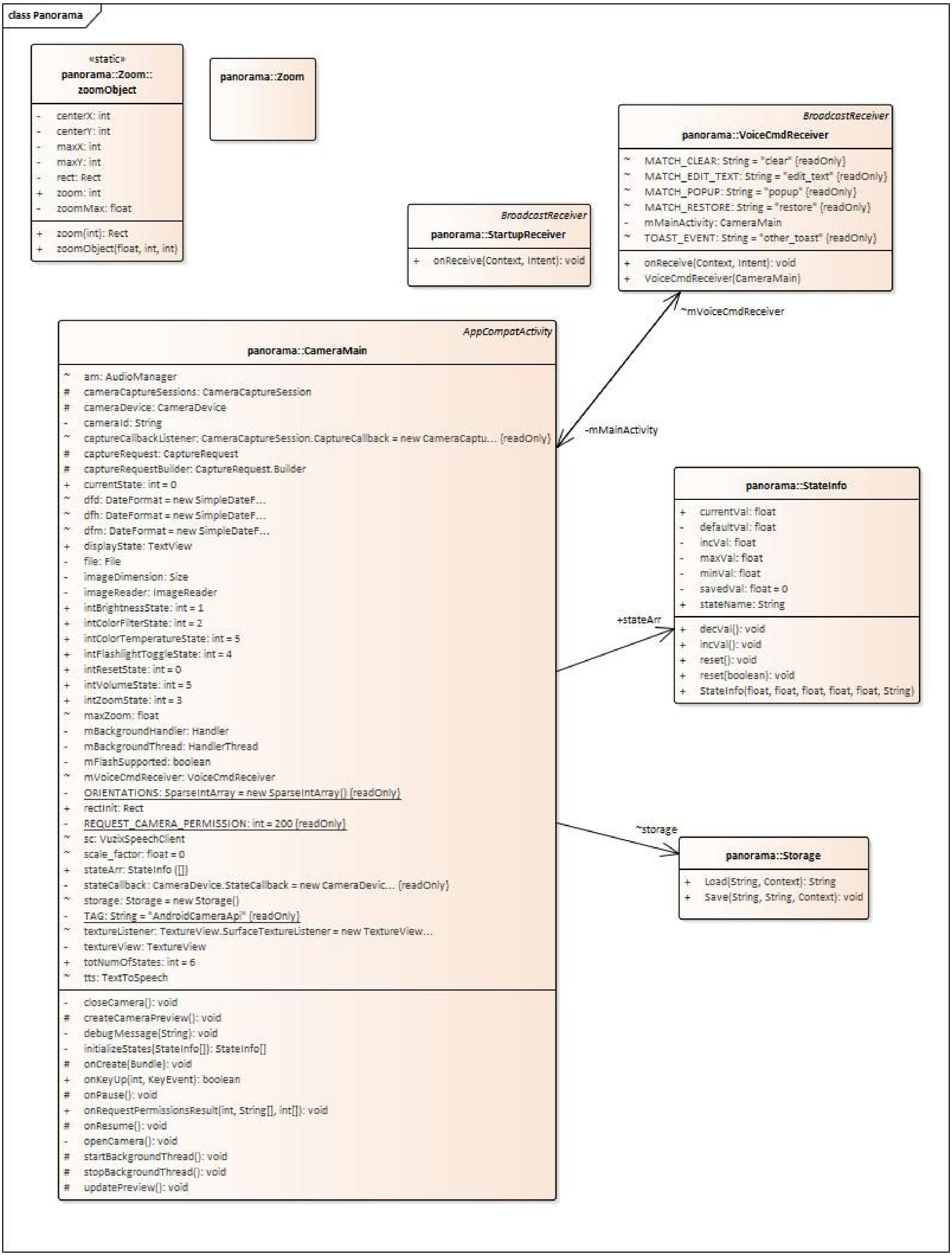

Figure 3.3:10 Class diagram

3.4 Validation & testing

Throughout the project we have had access to two examinees with relevant diseases in addition to several others having a shallow test of the product once upon a visit to Blindeforbundet. These tests with blindeforbundet were somewhat unstructured and were simply used to get a first impression of the participants using the glasses.

Internal product testing was constant throughout the project. In addition some more extensive testing was done upon request of project asigneer when looking into the use of the glasses in darker environments. This was done through a simple process of comparison between screens with or without the relevant features adjusted.

On the 6th of April a demonstration of the apps capabilities was done at one point with the patient with RP. The subject would walk the path from their bus stop to their office with and without the glasses.

The first extensive tests with a subject with retinal dystrophy was 19th of April 2018. The subject was given the glasses and asked to perform a series of basic tasks and give feedback of the feasibility of these for a person with their disease. A sheet of relevant questions were created beforehand and asked at the end of the session. These basic tasks mainly included attempting to read text and perceive detail in the environment. A testing component involving navigation was planned at one point, but was scrapped when it became obvious that the subject would have little to gain by using the app in this manner.

The final product test was done by giving the subject with RP the glasses with the app installed for the span of two days and record the feedback given afterwards. This was done in the period of May the 2nd to May the 4th. A sheet of questions was prepared and asked at the end of the session.

4 Discussion

This chapter will focus on how our solution worked in practice and the results of testing. What of our objectives we accomplished and how.

4.1 Feedback from tests

Some internal testing was performed on campus by group members in to investigate use in dim or dark environments on request of project assigner. We confirmed that adjusting screen brightness or the use of color filters did not impact how legible a dimly lit room was. This made sense as adjusting the brightness of the screen does not create more light for the camera.

We did however have some limited success with turning the glasses’ flash functionality on and using it as a flashlight. This allowed some darkvision, but its range was limited and the brightness of the light made looking directly at it somewhat uncomfortable for those around the wearer.

These two subjects were examinee 1 who has retinal dystrophy and examinee 2 who has Retinitis Pigmentosa.

4.1.1 Examinee 1

We met examinee 1 on the 19th of April 2018 for a more thorough survey how how well the glasses work for purposes not related to navigation. The test was to read text of varying size on a whiteboard and general usability of the glasses.

The result confirmed that using the glasses for viewing detail was too difficult to be practical for persons with weakened visual acuity. Their peripheral vision was still intact, so they therefore had little use for the increased field of view the glasses could provide.

The meeting did however lead to the discovery of a significant bug and a few pointers as to how to move forward with development. Examinee 1 expressed interest in OCR and a barcode reader, but these features were not implemented as of this meeting.

4.1.2 Examinee 2

Examinee 2 suffers from Retinitis Pigmentosa, was the most initially promising subject in superficial early testing and expressed the most interest in the product.

A bit more thorough testing was done during filming of a promotional video on behalf of project assigner. examinee 2 was asked to walk a specific route with or without the glasses and point out improvements to navigation. The subject seemed to find themselves glancing up at the screen in order to orient themselves within the environment and then proceeding onwards as they usually would.

Contradicting the results from our earlier tests, examinee 2 was able to use the screen to look around in environments that was otherwise too dim for them. Our earlier tests had apparently failed to consider just how limited a patient with RP's dark vision was.

Late in project development they were given the glasses over the span of a couple of days in order to do some more extensive testing in more realistic scenarios.
It should be mentioned that voice recognition was at that point broken due to a software update. The examinee used the glasses on trips to and from work, when exercising and when attending a church event.

As with most of those with RP they are fairly light sensitive. Due to the shape of glasses and the size of their sunglasses the subject was unable to wear both at the same time, the examinee was able to compensate by wearing a wide brimmed hat, but were still somewhat more light sensitive than usual.

Unfortunately the weather these two days was fairly sunny, the screen was therefore slightly more difficult to read on the subject's trip to work. Light from windows on the bus to and from work made searching for seats more difficult than it were in earlier tests,.

A feature which adjusted the screen's brightness was already implemented with the camera, but these changes were apparently not noticeable to the examinee at time of use. As per our internal tests, the glasses were of little use in darker environments. The subject mentioned that they had difficulties noticing cars in the distance. This may be caused by the limited size of the screen compared to the field of view from captured footage. Furthermore, the single camera caused a decreased sense of depth perception, objects close to the user appeared far more distant than they actually were.

The battery life of the glasses is remarkably brief which caused the subject some grief on a couple of occasions. Charging the glasses proved somewhat complicated as well as the charging port is located at a somewhat awkward spot on the battery. This issue would be compounded by the lack of sight by any potential users.

Equipping and unequipping the glasses was difficult with only one available hand or without any surfaces and similar. Again, in uncontrolled environments the glasses were less effective.
The examinee experienced similar issues at the church event as in the bus when light from windows made looking at the pews more difficult. The subject noted that use of the app was complex enough a that point in development. They also pointed out that several of the issues they encountered could be lessened as the examinee grew more accustomed to using the app.

Due to a lack of potential examinees the results from testing should be interpreted as nothing more than an indication rather than concrete fact.

4.2 Removed features

A state which changed color temperature was implemented at one point in development, but was removed due to lack of visible impact and clunky adjustments. The structure of this was planned to be used in order to implement more simple custom color filters. This would mostly just be a colored overlay across the screen, causing a slight tint in the display. Additional color filters implemented by this method was abandoned due to the filters making the screen less legible.

Picture taking was implemented through the example linked to in the references of this report. This feature was left out of later iterations of the app due to it requiring user input within the already limited user interface of the M300 glasses.Further, the feature was deemed as having little practical use by the individuals participating in testing.

We had ocr implemented through the use exsample we found,
it was under an open source licence apache v2 and built upon tesseract open source engine.
The example took the image feed from the camera and passed it through the tesseract ocr engine which had a trained ai that would decode the characters shown in the image.

The ocr had a see through rectangle on the screen where you would align the text inside that you would want to read. You could choose between two moduses where you could have it continues reading the text that was in the frame or you could take a picture.
The response time on the ocr was fast but we had a few problems with how the camera rotated while using it. Also you had to frame the text inside the box quite precisely for it to get a good read

on the text. This made the it difficult for a user to use it for our intended purpose. This ocr would also make our app crash at seemingly random times.
Maybe with some more development time or better hardware we could work some of the bugs out but at the current state we believe the app to be better without.

4.3 Problems encountered during development

We had a problem with one of the updates that broke our code for the speech recognition. We spent several days to fix this. The bug appeared in the creation of the voice recognition object. The fix was using an example from the the vuzix developer center that instigated the voice recognition differently.

During development we only had access to a single set of M300 glasses which slowed development down significantly. While working on the project we often had to take turns passing the M300 glasses around to test out different features. The emulators in android studio was often not sufficient enough to test on.

Early in project development we experienced some issues in which the app would crash after a certain amount of time had passes. We eventually realized this issue was caused by the glasse's sleep mode function. The problem was fixed by simply setting a flag in onCreate which removed sleep mode from the app entirely.

```
getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
```

The documentation on Vuzix developer center section regarding interaction methods was incorrect when listing the key code for a long press of the backmost button. By printing the keyCode variable within the onKeyUp() function we discovered that the actual keycode was Keycode_Back with a value of 4. Vuzix seems to have corrected this error at the time of writing this report.

We had problems implementing the Vuzix M300 sdk as setting the project to compile to it lead to a flurry of errors. We had to delete several pages of code that where implemented in a newer Android SDK and then we had to force the compiler to use an older build tool this was necessary because there were some features only accessible through the m300 sdk.

The connecting cable between the glasses and the integrated battery was somewhat fragile at project start, eventually one of the cable locks broke apart. We were able to temporarily fix the issue by plastering the battery in duct tape and permanently attaching the cable to the battery. This worked, but we still experienced some minor issues during testing in which the glasses lost connection to development laptop or loss of power entirely. As stated earlier this was simply a temporary solution, ideally we would have purchased a replacement cable, but this was never a priority.

Figure 4.3:1 Damaged wire

4.3.1 Lingering issues
Combining the AR glasses with regular glasses and sunglasses. Many people suffering from eye condition are also sensitive to light. They therefore have to wear sunglasses when outside and the smart glasses has some trouble getting over the sunglasses. In addition the screen is hard to see through the sun glasses. Note that sunglasses designed for light sensitive persons are often bulkier than sunglasses worn by persons with healthier sight in order to eliminate as much unnecessary peripheral light as possible. This issue can be lessened through use of anything which shades the user’s eyes such as a wide brimmed hat, but this will still not eliminate the issue of peripheral light outside of ceiling lamps or sunshine.

After extensive testing we discovered that the Vuzix M300 camera may not support high dynamic range rendering which is necessary to manipulate tonemaps and functionally implement planned features such as contrast and more elaborate color filters.

The lifetime of the battery is only 2 hours per charge with the on-board battery. So for prolonged use it is necessary to have an external battery to power up the device. There is also the issue with how complicated connecting the charger to the device is for a person with impaired vision.

Some of our examinees had issues with learning how to charge the device, the two parted structure and usb connection may be an issue with the less technologically literate section of the population. This issues is only worsened by limited vision of users.

Then there is the matter of how our solution would potentially be sold and distributed. The glasses are expensive, perhaps prohibitively so, at 1699€ [14]. There are, however, alternatives to selling it directly to consumers. Hjelpemiddelsentralen and other similar agencies can foot the bill and aid in delivery of the product.

5 Conclusion

From our attempt to create an app for smart glasses which relieved some of the symptoms for some sight related diseases we have received mixed results.

Trials involving those with diseases like MD had little use of our solution due to lack of visual acuity. Regardless of how detailed or adapted the screen is, it makes little difference when it isn't legible at all to the user. It should be noted however that usability may vary from person to person and testing was not extensive enough to eliminate individual factors or variance in the expression of the disease.

Tests involving those suffering from RP were initially promising. The subjects seemed to be able to find their bearings in environments easier and could use the glasses in lower light environments than they usually would be able to. These results were found in controlled situations in which the examinees had helpers around them to aid in equipping and unequipping the glasses and explain how the app functioned.

When the subject was given the glasses over a period of several days, they encountered several issues with the specifications of the glasses, lighting and usability. While this didn't eliminate the boons from using the glasses entirely and several of these problems may be resolved through simple practice with using the glasses. Some problems are more intrinsic within the glasses' current hardware capabilities and the extent of the app.

In the future in similar projects aimed at this same group of people, it may be prudent to implement more extreme progressive lighting changes than would normally be found in supported cameras. This hypothetical lighting feature should also have some cap for how intense the light may get.

Facial recognition was a requested feature not looked into at all by this project due to limited development time, but something similar done correctly could be a huge boon to its targeted users. A more adapted OCR and a barcode scanner may be useful to some subjects, but be cautious of unnecessary user interface complexity.

In summary this field holds potential for being a good tool for people with ocular disabilities. With time we hope that the capabilities of hardware improves, giving potential developers access to higher resolution screens, wider camera field of view and longer battery life.

6 References

[1] Nei.nih.gov. “Facts About Retinitis Pigmentosa”. (2018). [online] Available at: https://nei.nih.gov/health/pigmentosa/pigmentosa_facts. [Accessed 15 Feb. 2018].

[2] Nei.nih.gov. “Facts About Age-Related Macular Degeneration”. (2018). [online] Available at: https://nei.nih.gov/health/maculardegen/armd_facts. [Accessed 15 Feb. 2018].

[3] This project’s preliminary report. This should be found within an appendix.

[4] eSighteyewear.com. ”eSight - Electronic Glasses for the Legally Blind”. (2018). [online] Available at: https://www.esighteyewear.com/ [Accessed 11 May 2018]

[5] eSighteyewear.com. “eSight FAQ - Electronic Glasses for the Legally Blind”. (2018). [online] Available at: https://www.esighteyewear.com/faq#Who-does-eSight-work-for [Accessed 11 May 2018]

[6] IrisVision.com. “IrisVision Low Vision Aids | See Clearly, Live Fully”. (2018). [online] Available at: https://irisvision.com/ [Accessed 11 May 2018]

[7] Orcam. “OrCam MyEye 2.0”. (2018). [online] Available at: https://www.orcam.com/en/myeye2/ [Accessed 11 May 2018]

[8] CyberTimez.com. “Cyber Eyez M300 Complete Package – Cyber Timez — Makers of Cyber Eyez”. (2018). [online] Available at: https://cybertimez.com/?product=cyber-eyez-M300-complete-package. [Accessed 11 May 2018]

[9] NuEyes.com. “NuEyes Pro – NuEyes”. (2018). [online] Available at: https://nueyes.com/nueyes-pro/ [Accessed 11 May 2018]

[10] Samsung.com. “Samsung Gear VR with Controller - The Official Samsung Galaxy Site”. (2018). [online] Available at: https://www.samsung.com/global/galaxy/gear-vr/ [Accessed 11 May 2018]

[11] StarTrek.com. “Star Trek La Forge, Geordi”. (2018). [online] Available at: http://www.startrek.com/database_article/la-forge-Geordi [Accessed 11 May 2018]

[12] Vuzix.com. “M100 Smart Glasses Hands-Free Mobile Computing”. (2018). [online] Available at: http://files.vuzix.com/Content/pdfs/Vuzix-M100-Product-Sheet-01-01-2016.pdf [Accessed 11 May 2018]

[13] Vuzix.com. “M300 Smart Glasses Hands-Free Mobile Computing”. (2018). [online] Available at: http://files.vuzix.com/Content/docs/north-american/web/Vuzix-M300-Product-Sheet-01-18.pdf [Accessed_ 11 May 2018]

[14] Vuzix.com. “Vuzix M300 Augmented Reality (AR) Smart Glasses”. (2018). [online] Available at: https://www.vuzix.com/Products/M300-smart-glasses [Accessed: 11 May 2018]

[15] OsterhoutGroup.com. “ODG - R-7 Smartglasses”. (2018). [online] Available at: https://www.osterhoutgroup.com/r-7-smartglasses [Accessed: 11 May 2018]

[16] AdsReality.com. “A brief history of augmented reality - AdsReality”. (2018). [online] Available at: http://adsreality.com/history-of-augmented-reality-infographic/ [Accessed: 11 May 2018]

[17] L.B. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in Proceedings of IEEE Virtual Reality Annual International Symposium, 18-22 Sept., 1993, Seattle, WA, USA [PDF]. Stanford, CA, USA: Center for Design Research, Stanford University, 1993. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=380795. [Accessed: 11 May 2018].

[18] developer.android.com. “Android Developers”. (2018). [online] Available at: https://developer.android.com/. [Accessed: 12 May 2018]

[19] Vuzix.eu. “Getting started with M300 Development”. (2018). [Online] Available at:https://www.vuzix.eu/Developer/Dashboard/M300. [Accessed: 12 May 2018]

[20] youtube.com. “The Blind Life”. (2018). [online] Available at: https://www.youtube.com/channel/UCNbzN3eHbLKPzltSB560DkA. [Accessed: 12 May 2018]

[21] esighteyewear.com. ”eSight FAQ - Electronic Glasses for the Legally Blind”. (2018). [online] Available at: https://www.esighteyewear.com/faq#How-do-I-find-out-more-about-getting-eSight-3-and-if-I-can-afford-it. [Accessed: 12 May 2018]

[22] businessinsider.com. “An insider's look at the tumultuous launch of Google Glass”. (2015). [online] Available at: http://www.businessinsider.com/google-glass-launch-2015-2?r=US&IR=T&IR=T. [Accessed: 13 May 2018]

[23] Vuzix. “Vuzix M300 Smart Glasses and New Line of Accessories Now Available for Public Purchase ” (2017). [online] Available at: https://d2iankuf53zudv.cloudfront.net/Content/Upload/PDFs/06_13_2017_M300_Website_and_Accessories_Fin.pd. [Accessed: 13 May 2018]

Code References:

camera2 example:

[24] inducesmile.com. “Android Camera2 API Example Tutorial” (2016). [online] Available at: https://inducesmile.com/android/android-camera2-api-example-tutorial/ [Accessed: 11 May 2018]
Received permission from author

Zoom example:

[25] stackoverflow.com. “java - Zoom Camera2 Preview using TextureView - Stack Overflow” (2015). [Online] Available at: https://stackoverflow.com/questions/32711975/zoom-camera2-preview-using-textureview [Accessed: 10 May 2018]

Start on boot:

[26] stacktips.com. “How to Start an Application at Device Bootup in Android” (2013). [Online] Available at: http://stacktips.com/tutorials/android/how-to-start-an-application-at-device-bootup-in-android. [Accessed: 11 May 2018]

Save files example:

[27] developers.android.com. “Save files on device storage”(2018) [Online] Available at: https://developer.android.com/training/data-storage/files
[Accessed: 14 May 2018]

7 Appendices

Appendix list:

Appendix A	Abbreviations
Appendix B	User manual
Appendix C	Project plan
Appendix D	Testing summary
Appendix E	Preliminary report

Appendix A Abbreviations

(A)MD:	(Age related) Macular Degeneration
API:	Application Programming Interface
AR:	Augmented Reality
FOV:	Field Of View (measured in degrees)
HUD:	Heads Up Display
RP:	Retinitis Pigmentosa
SDK:	Software Development Kit
VR:	Virtual Reality

Appendix B User manual

M300

The glasses has four buttons: ON/OFF, left arrow key, right arrow key, and enter/confirm

Press the arrow keys to navigate around in the main menu. Push ENTER to start the app in the middle of the screen.

Pressing and holding the middle key will exit the entered application.

The touch plate works as an alternative to the arrow keys

Panorama

The app has six modes one can iterate through. Forwards and backwards. Do this by pressing and holding the frontmost or middle button, the arrow keys. A voice will tell you what mode you have switched to.

1. Reset: Resets all modes to standard settings.
2. Brightness: Changes screen brightness.
3. Color Filter: selects between color filters.
4. Zoom: Zooms in and out.
5. Flashlight: Toggles flashlight on and off.
6. Volume: Adjusts volume of voice.

In the selected mode, a single press of the middle or frontmost button will adjust the value/setting of that mode. While a press of the rearmost button, the enter key, will alternate the mode between the standard value and the last selected value.

Pressing and holding middle key will exit Panorama.

N.B. the enter key is only used to confirm/enter a selection in the glasses’s main menu, NOT in Panorama.

Appendix C Project plan

Project title: Adapting smart glasses for persons with loss of peripheral vision
The project was assigned to us by Dr. Harald Reiso of Sørlandet Hospital. Work was distributed equally between group members, each of us taking tasks we felt we could contribute most by doing at that particular time. Jira was used in order to organize the different tasks and to track hours. The group consisted of the following members: Jonas Claesson, Morten Grundetjern, Audun Borgersen
The tasks completed by each group member:

Morten Grundetjern:
Implemented base camera (partially due to a miscommunication two of the group members created the base camera functionality independently of each other)
implemented Brightness adjustments
implemented zoom
implement Speech recognition
implemented volume controls
implemented tts text to speech
storage made it possible to save and load
preliminary report writing
report writing
bug fixing
Had implemented ocr but it made the app unstable so we decided to leave it out

Jonas Claesson:
- Worked on contrast before it was abandoned
- Report writing, grammar checking
- Speech recognition
- Wrote user manual
- General testing of software
- Worked on OCR, testing
- Testing and modifying text-to-speech
- Bug fixing
- Cleanup

Audun Borgersen:

implemented base camera (partially, due to a miscommunication two of the group members created the base camera functionality independently of each other)

created menu and state system

implemented color filters, base camera settings

implemented flashlight functionality

implemented boot directly to app

created and removed color temperature and additional color filter system

preliminary report writing

report writing

looked into several possible contrast solutions

bug fixing

cleaning code

Project : DAT304G18V - Medisinsk AR App for Synshemmede (MEDSYNAPP) | January 29, 2018 - March 31, 2018

Name	P	%	Σ	29 M	30 T	31 W	01 T	02 F	03 S	04 S	05 M	06 T	07 W	08 T	09 F	10 S	11 S	12 M	13 T	14 W	15 T	16 F	17 S	18 S	19 M	20 T	21 W	22 T	23 F	24 S	25 S
Jonas Claesson		100%	103		4.5	5	4	5.5				4.5		2	4.5				3.5	6.5	2	4			4				4		
Karl Audun Kagnes Borgersen		100%	124		4.5	5	5	5.5				4.5		2	4				3.5	8.5	2	4			4		6	2			
Morten Grundetjern		100%	119				4	5.5							13.5				3.5	7.5	2	4			4		6				
Daily hours total:					9	10	13	16.5				9		4	22				10.5	22.5	6	12			12		12	2	4		
Weekly hours total:									48.5							35							51							30	
Planned hours total:																															

Name	26 M	27 T	28 W	01 T	02 F	03 S	04 S	05 M	06 T	07 W	08 T	09 F	10 S	11 S	12 M	13 T	14 W	15 T	16 F	17 S	18 S	19 M	20 T	21 W	22 T	23 F	24 S	25 S	26 M	27 T	28 W	29 T	30 F
Jonas Claesson					6				3	5	5	5			4	3	5	3				2	3	5									
Karl Audun Kagnes Borgersen					6				3	8	6	6				8.5	5	2	4			4	4	6		1							
Morten Grundetjern					6			2	6	5	6	6			6	6	6		2			6		12									
Daily hours total:					18			2	12	18	17	17			10	17.5	16	5	6			12	7	23		1							
Weekly hours total:						18							66							54.5							43						
Planned hours total:																																	

Total	To date	Period
Worked	346	346
Planned	0	0

Project : DAT304G18V - Medisinsk AR App for Synshemmede (MEDSYNAPP) | April 1, 2018 - May 13, 2018

Name	P	%	Σ	01 S	02 M	03 T	04 W	05 T	06 F	07 S	08 S	09 M	10 T	11 W	12 T	13 F	14 S	15 S	16 M	17 T	18 W	19 T	20 F	21 S
Jonas Claesson		100%	103.5																		5	1		
Karl Audun Kagnes Borgersen		100%	111																		5	3	2	
Morten Grundetjern		100%	125.5																	6	5	3	2	
Daily hours total:																				6	15	7	4	
Weekly hours total:																								32
Planned hours total:																								

Name	22 S	23 M	24 T	25 W	26 T	27 F	28 S	29 S	30 M	01 T	02 W	03 T	04 F	05 S	06 S	07 M	08 T	09 W	10 T	11 F	12 S	13 S
Jonas Claesson		7.5		5	5	4			6	5	3	4	7		3	4	7	8	7	6	6	10
Karl Audun Kagnes Borgersen		6		4					15	6	6	6	10			6	8	7	6	6	6	9
Morten Grundetjern		6.5	6	4	6	6			6	6	6	7	8			6	8	6	6	6	6	10
Daily hours total:		20	6	13	11	10			27	17	15	17	25		3	16	23	21	19	18	18	29
Weekly hours total:							60							101							118	29
Planned hours total:																						

Total	To date	Period
Worked	340	340
Planned	0	0

Appendix D Testing summary

Examinee 1:

Met Olav Nielsen on the 19th of April 2018. Olav suffers from retinal dystrophy, but can navigate just fine by himself as his peripheral vision is intact, he does however have trouble focusing on detailed objects. He is a little light sensitive, wearing a capped hat made looking at the screen less unpleasant. Reading through the screen was difficult without bold letters. Would be interested in an OCR feature, buss timetable, barcode scanner.

A flaw was discovered when the camera failed to adjust properly when looking through the left eye. He had some difficulty distinguishing color. As his vision focus is lacking he gained little in terms of detail from the glasses screen. There does not seem to be much overlap between the needs of those suffering from retinal dystrophy and those with Retinitis Pigmentosa.

Examinee 2:

An extensive test was performed in which examinee 2 was given the glasses with the app installed for the duration of 2 days. Examinee 2 suffers from Retinitis Pigmentosa and was the most promising subject for our application. We were given a fair amount of useful feedback, though the use of the glasses proved difficult for the examinee. The subject is fairly light sensitive which caused some issues, but this is common for such diseases. An update from Vuzix had broken voice recognition at the time of testing and was therefore unfortunately.

The glasses were difficult to use without others around to help the examinee. Use when carrying things was difficult.

Adjusting placement of glasses for each use was tedious.

The glasses had to be a certain distance from the examinee which proved difficult to maintain.

Judging distance was difficult.

The subject had difficulty adjusting brightness to a comfortable level due to variance in natural light.

The examinee usually wears sunglasses, but since they are too wide to wear with the glasses the examinee instead opted to wear a wide brimmed hat instead. This blocked sunlight, but other light was still an annoyance.

The glasses has trouble making out objects in the distance.

To have the glasses boot directly into the app may be useful.

The examinee had some issues charging the hardware.

There may be a bug in which the glasses enter sleep mode within the app for some reason.

The glasses can not be used in darker environments.

Light from windows such as those lining the pews of a church was an annoyance.

The app was complex enough at this point, adding additional features may be pointless.

Some of these issues is a simple matter of getting accustomed to the hardware, however many of the subject's comments can be useful adjustments to make the app more user friendly.

Brightness was adjusted a fair amount.

Color filter was not used at all.

Zoom was not used at all.

Flashlight was not used at all.

Volume was adjusted initially.

Appendix E Preliminary report

See separate submission.