Title of Thesis

Place your subheading here

Description of thesis (semester- / Bachelor thesis / etc.)

[Insert short text (abstract) if desired]
This document serves as a template for the compilation of reports according to the guidelines of the BFH. The template is written in LATEX and supports the automatic writing of various directories, references, indexing and glossaries. This small text is a summary of this document with a length of 4 to max. 8 lines.
The cover picture may be turned on or off in the lines 157/158 of the file template.tex.

Degree course:	[z.B. Electrical and Communication Engineering]
Authors:	[Test Peter, Müster Rösä]
Tutor:	[Dr. Xxxx Xxxx, Dr. Yyyy Yyyy]
Constituent:	[Wwwww AG]
Experts:	[Dr. Zzzz Zzzz]
Date:	07.02.2014

Versions

Version	Date	Status	Remarks
0.1	01.08.2013	Draft	Lorem ipsum dolor sit amet
0.2	21.08.2013	Draft	Phasellus scelerisque
0.3	02.09.2013	Draft	Donec eget aliquam urna. Lorem ipsum dolor sit amet
1.0	26.01.2014	Final	Lorem ipsum dolor sit ametPhasellus scelerisque, leo sed iaculis ornare
1.1	31.01.2014	Correction	Layout changed
1.2	07.02.2014	Addition	Chapter 1.1 extended

Management Summary

Contents

1. Introduction

1.1. Eye Tracking

Eye-tracking is utilised to monitor eye movement. It is employed in a wide variety of fields such as marketing research, psychology, virtual reality and sports. In sport research eye-tracking offers important insights about the hand-eye coordination or visual search techniques. These differ greatly between professional athletes and beginners so the analysis can improve the efficiency of training.

An eye-tracking device for sports needs a high accuracy and temporal resolution to draw accurate conclusions. As the device should interfere as little as possible with the activity of an athlete, it should also be portable.

1.2. Current State

In collaboration with the sport research institute at the University of Bern, the microLab research group has developed an eye-tracking system for sports. This system utilizes two cameras per eye to capture infrared images at a high speed of 120 frames per second. The information from the cameras should be used to determine the gaze direction of the athlete. An important part of that is the detection of the pupil, which is not implemented satisfactory yet.

1.3. Overview of the Eye-tracking System

The Gazelle eye-tracking system consists of two main parts. The first is the GazelleGlasses where the cameras are mounted and the data is sent down to the second main part which is GazelleCompute. As the name suggests this component is responsible for the processing of the data.

GazelleGlasses

For the capture of an eye two cameras are placed below the eye and besides the nose. This positioning was chosen to not obstruct the view of the athlete and offer a good perspective on the eye. A front facing camera is placed between the eyes and provides a view that is close to the view of the athlete.

The data that is generated by the cameras needs to be transferred to GazelleCompute. This is done by serializing the data and sending it over an Ethernet cable to the main compute unit.

GazelleCompute

The main processing is done on a small portable unit. The units main components are a Zynq FPGA and a Tegra 3. Initially the idea was to do the image processing on the Tegra 3 as it is a powerfull mobile System on Chip.

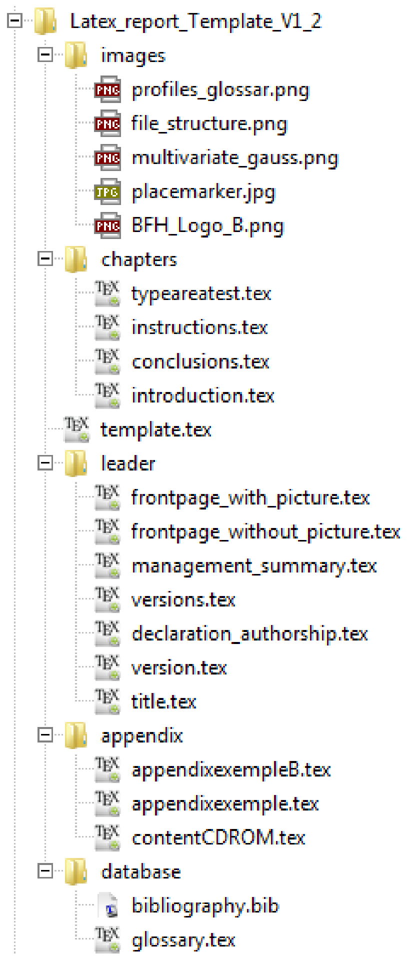
Problems with the data transfer from the FPGA to the Tegra 3 scrambled that idea. The image processing is now required to run on the dual core arm CPU that is on the Zynq FPGA.

1.4. Pupil Detection

1.5. Organising Documents

This document is structured according to the documentation of a project work or a thesis. In Chapter 2, the packages used are briefly explained, and instructions are given, how the bibliography and the glossary are to be used. Chapter **??** presents a sample chapter to audit the type area.

In figure 1.1 the file structure is shown for this template.



```
Latex_report_Template_V1_2
    images
        profiles_glossar.png
        file_structure.png
        multivariate_gauss.png
        placemarker.jpg
        BFH_Logo_B.png
    chapters
        typeareatest.tex
        instructions.tex
        conclusions.tex
        introduction.tex
    template.tex
    leader
        frontpage_with_picture.tex
        frontpage_without_picture.tex
        management_summary.tex
        versions.tex
        declaration_authorship.tex
        version.tex
        title.tex
    appendix
        appendixexempleB.tex
        appendixexemple.tex
        contentCDROM.tex
    database
        bibliography.bib
        glossary.tex
```

Figure 1.1.: File structure

1.6. Contact

The manufacturers of this template welcome any suggestions for improvement. Chapter 1.7 shows possible suggestions for improvement.

First Name Last Name	**E-mail**	**Function**
Alfred Kaufmann	alfred.kaufmann@bfh.ch	Employer, Project Management, Supplements, Improvements
Fritz Dellsperger	Retired	Tips on the structure and layout
David Burri	Contracted out	First compilation of the Template

Table 1.1.: Contact Persons

1.7. Suggestions for Improvement

- Create a BFH Style Files
- Template for the Compilation of presentations with LaTeX

2. Instructions

The following table shows some of the most important packages used in the LaTeX template.

Package	Function
`cmbright`	sans serif font "Computer Modern Bright", which supports text encodings OT1, T1 and TS1, as well as the mathematical signs and AMS symbols
`ae`	provides better resolution fonts in PDF files
`fancyhdr`	easy adjustment of head- and foot lines
`graphicx`	integration of graphics in LaTeX documents
`booktabs`	better presentation of tables
`textpos`	simplified and absolute positioning of boxes on the page
`hyperref`	package to complie links into PDF files
`geometry`	simplified and improved adaptation of the standard type area
`makeidx`	simple Index compilation (see section 2.1)
`glossaries`	compilation of glossaries (see section 2.2)

Table 2.1.: Packages

2.1. Subject Indices

LaTeX is not able to create an Index in the basic configuration. This can be created in LaTeX with the `makeidx` package and the `makeindex` program. The following page contains a detailed explanation of how the package works, and its application:

`http://en.wikibooks.org/wiki/LaTeX/Indexing`

Roughly summarized the following points are needed for an index:

- Embed the package `makeidx`.
- Initialize the compilation with the command `\makeindex`.
- Continuously initializing words in the text with the command `\index{}`.
- During the first passage of the document's compilation, the directory is created and definitions marked with `\index{}` are stored in the `.idx` file.
- During the second passage the `.idx` file is sorted, formatted and stored as `.ind` file whereas LaTeX then inserts the `.ind` file into the document.

2.2. Glossay

A glossary can also be created in LATEX with the `makeindex` program and the `glossaries` package. The following list shows the procedure to generate a glossary:

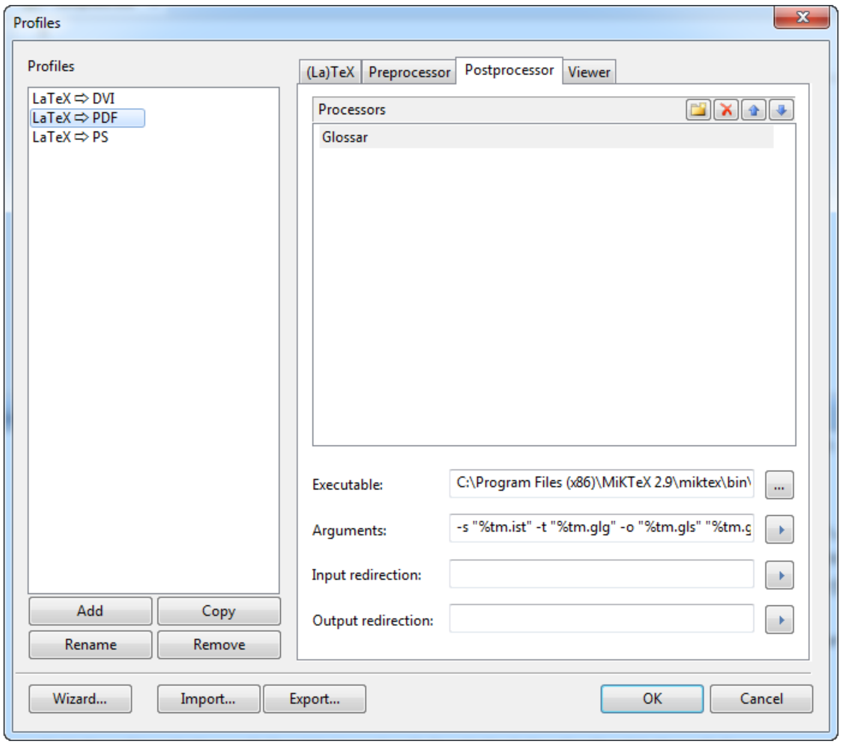
- Integration of the package `glossaries`.
- If necessary, a personal database may be created including glossary entries. This template works with such a database, which is stored in the `databasefolder`. Entries from the database are only written in the directory if the word in the text is actually stated.
- With the `\makeglossaries` command a new compilation is initialized.
- New entries can be created with the command `\newglossaryentry{<SHORTCUT>}{name={<NAME>},description={<DESCRIPTION>}}`.
- In the text continuously referencing words with the command `\gls{<SHORTCUT>}`.
- Similar to the compilation of the index, the directory is only embedded into the document during the second passage.

In order to work accurately, the glossary must be compiled with `makeindex` after post-editing the document. For this the following code in the command line is to be executed:

```
makeindex -s template.ist -t template.glg -o template.gls template.glo
```

With most LATEXeditors, this can be stated as a post-processing step. The following explanation is for the TeXnic-Center program. Under the menu "Build" > "Define Output Profile..." (short: alt + F7) in the "Postprocessor" register, the window shown in Figure 2.1 can be found. Then it is necessary to insert a new entry, when an application as well as an argument must be specified. The application can be found in the MiKTeX installation (`..\MiKTeX X.X\miktex\bin\makeindex.exe`). As an argument, the following line must be entered:

```
-s "%tm.ist" -t "%tm.glg" -o "%tm.gls" "%tm.glo"
```



Executable: C:\Program Files (x86)\MiKTeX 2.9\miktex\bin\
Arguments: -s "%tm.ist" -t "%tm.glg" -o "%tm.gls" "%tm.g
Input redirection:
Output redirection:

Figure 2.1.: Post-processing

2.3. Bibliography

To compile a bibliography one must resort to BibTeX. The folder `database` includes a `.bib` file with various database entries. How the entries are to be compiled, can be taken from various sources of the Internet or books. The entries in the database will only be written to the directory of the document when the source is actually cited in the text.

Under the following addresses further explanations are found in order to compile the database and its use:

- `http://en.wikipedia.org/wiki/BibTeX`
- `http://www.bibtex.org/`

3. Model

A very important step in hardware and software development is testing. For an eye tracking system it requires a big effort to do this with real data. The reason for that is that the information where the participant looks at each moment needs to be recorded alongside of the videostreams. For this project real time data processing is needed as there is no hardware available on Gazelle Compute to encode or store the video streams of the cameras.

To optimize the algorithms a simple way of generating new data where the gaze direction and the camera positions are known would simplify the process.

3.1. 3D-Modeling Software

The 3D-Modeling software to create the model needs to fulfill a few requirements:

- Correct representation of refracting light
- Create animations
- Be scriptable
  - Set position and rotation of certain objects at a certain frame
  - Set camera properties and which camera is active
  - Read the position and rotation of objects at any frame
- (optional)Be free
- (optional)Run on Linux
  - Server that can be used to render runs Linux containers
- (optional)Be easy to learn

Blender is able to match all requirements although the last one might be debatable. You can apply a material properties to a surface. This includes refracting. Rotation and position of objects can be inserted as keyframes and it interpolates the steps in-between.

Blender is also an open source project that is built with Python and has an powerful API that gives access to nearly everything.

3.2. Requirements for the Model

In order to create a flexible model that is close to the reality, the model needs to follow the following requirements:

- Correctly represent a human eye
- Accommodate for different eye parameters
  - Eye diameter
  - Distance between Eyes
- Correctly represent camera placement and rotation
- Correctly represent camera properties
  - Resolution

  - Framerate
  - Field of View
- Gaze direction can be animated
- Classes rotation and position can be animated
- Front facing cameras records gaze direction

3.3. Human Eye

A side view of a human eye model is visible in 3.1. The camera can only see the front part of the eye. As a result only the cornea, anterior chamber and the iris need to be modeled correctly. The lens can be simplified by a black surface.

It might be counterintuitive that the cornea extends as much out of the eyeball as visible in 3.1. An easy way to verify is to close the eye and move the eyes while holding a finger on the eyelid.

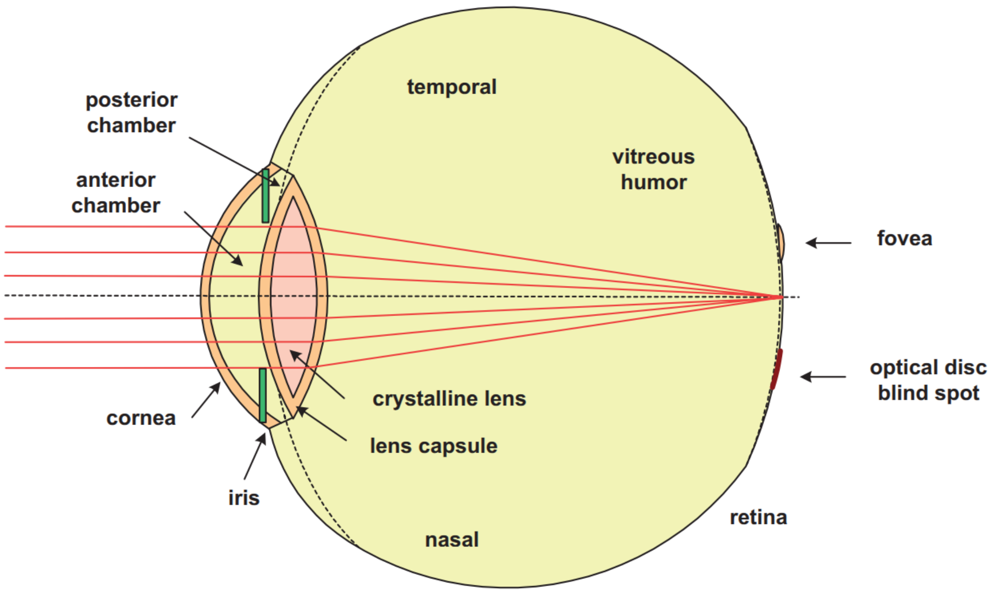


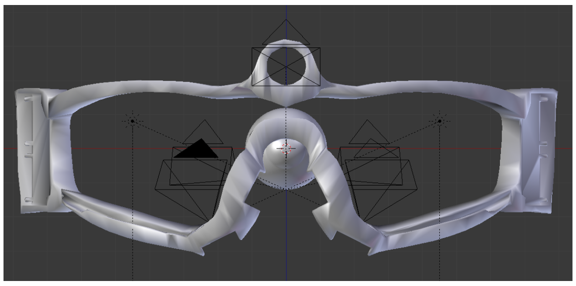
Figure 3.1.: Side view of a human eye model [1]

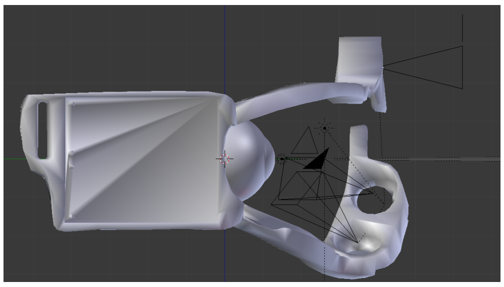
There are a few models with different values for the refracting indexes, radii and distances but they are close together in most cases. So the simplified Gullstrand eye was chosen for the model.

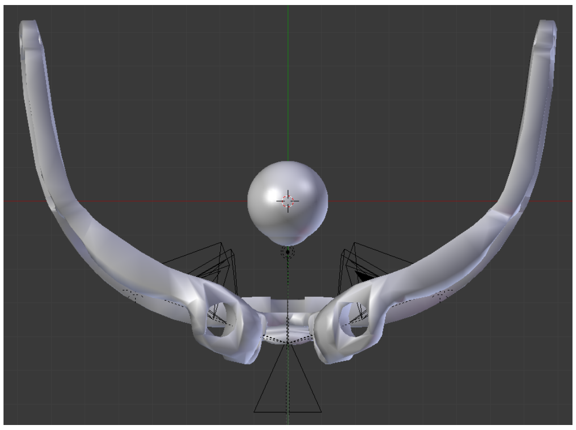
Notaton	Relaxed			Accomodated		
	Radius r [mm]	Thickness d [mm]	Index n	Radius r [mm]	Thickness d [mm]	Index n
cornea	7.70	0.50	1.376	7.70	0.50	1.376
anterior chamber	6.80	3.10	1.336	6.80	2.70	1.336
crystalline lens	10.0	3.60	1.4085	5.33	4.0	1.426
vitreous humor p	-6.00	17.187	1.336	-5.33	13.816	1.336

Table 3.1.: Properties of the eye with the simplified Gullstrand eye. [1]

A human eye has on average a diameter of 24mm and the distance between the eyes ranges from 56mm to 72mm with the mean for men at 65mm and 62.6mm for women. [1]


(a) Front view of base model


(b) Side view of base model


(c) Bottom view of base model

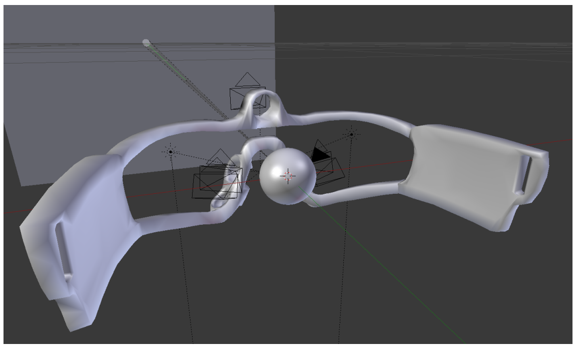

(d) 3D-view of base model that shows also the light cone that follows the eye rotation

Figure 3.2.: Finished base model with eyes in the center. A Script will place the eye correctly and generate an animation from the model that is shown here.

3.4. Base Model

While there are quite a few parameters that need to be adjusted, some will remain the same. Those can be incorporated into a base model that can then be adjusted by a script for each render.

This base model is centred around an accurate eye model that is built with the properties mentioned in section 3.3. Every eye is a group that can be manipulated with a script as one object. This includes a directed light cone that points where the eye is looking.

An existing model of the glasses is used and placed to approximately represent the position how they would be in the real world. Cutouts for the cameras are used to place them correctly while the rotation is approximated to capture an average positioned eye correctly. The light cones that represent the gaze direction illuminate a small portion of a wall that is captured by the front facing camera and can be seen in figure 3.2d. As with the eye, the cameras and the glasses build a single object that can be moved and rotated. The result can be seen in figure 3.2

3.5. Scripting

The base model alone is not enough to create meaningful results because there is no animation, the eyes are not placed and blender would just render one camera. Additionally the position and rotation of the eye and glasses needs to be logged for every frame as they are required for comparing the final gaze direction results. To allow a wide variaty of animations to be created a simple way of configuring the script is needed.

Automate Animation

This script is written in Python because Blender offers a powerfull Phython API that allows such a script to modify every property in a model. JSON is a simple data-interchange format that is easy to read and write for humans. Sites such as `http://jsoneditoronline.org/` offer a convenient way to modify JSON. The script uses a JSON file with all the properties as input and adjusts the values in the model and saves a modified blend file for each camera that is ready to be rendered. The properties includes static parameters such as the eye position and diameter and the number of frames that should be rendered. But also dynamic information for keyframes such as the rotation of the eye and position and rotation of the glasses.

Automate Process

With the script the process to generate render output is as follows:

```
#Use script to generate render ready files
blender base.blend -b -P generate.py -- animation.json
#render the animation for each camera
blender -b leftDown.blend -E CYCLES -t 4 -a
blender -b leftUp.blend -E CYCLES -t 4 -a
blender -b rightDown.blend -E CYCLES -t 4 -a
blender -b rightUp.blend -E CYCLES -t 4 -a
blender -b scene.blend -E CYCLES -t 4 -a
```

This is still very time consuming and can be automated further.

A Makefile offers a convenient way to automate the stages further and make it simple to clean the whole mess up once the generated or rendered files are not needed anymore. With the Makefile the whole process is shorter and easier to remember:

```
#export config file
export ANIMATION_JSON=animation.json
make
make render
```

An added benefit is also that make will start the render jobs detached and distribute the available cpu cores equally among the render jobs.

4. Pupil Detection

Without test data it is very hard to improve an algorithm. The generated pictures of the last chapter lay the groundwork to detect the pupil digitally.

This chapter shows how the current algorithm tries to detect a pupil. Improvements are presented that achieve a better result.

4.1. Overview of Existing Algorithm

The general sequence of the algorithm can be described as follows:

1. Find darkest spot
2. Extract rays starting from the darkest spot from the image
3. Transit the rays with a filter of certain length to detect edges
4. Fit an ellipse on the detected edges with the least square method

The darkest spot is determined by a raster with a fixed distance between points. For each point the average with the surrounding eights point is calculated. The point with the darkest average is the startpoint for the rest of the algorithm.

Outgoing from the startpoint rays are stamped out by the starburst IP that is described in "Eye Tracking for Sports, Chapter 18.5"

The difference to the next pixel is calculated for each ray that is stamped out by the starburst IP. Because the edge may not lie on a single pixel surrounding differences are added together to build a score for each pixel. This score is positive for transitions from a darker pixel to a brighter. The pixel with the highest score is determined as detected edge. So only edges that transit from dark to bright can be detected.

Every edgepoint is handed over to a weighted least square method to find a ellipse that fits all the points well.

Testing

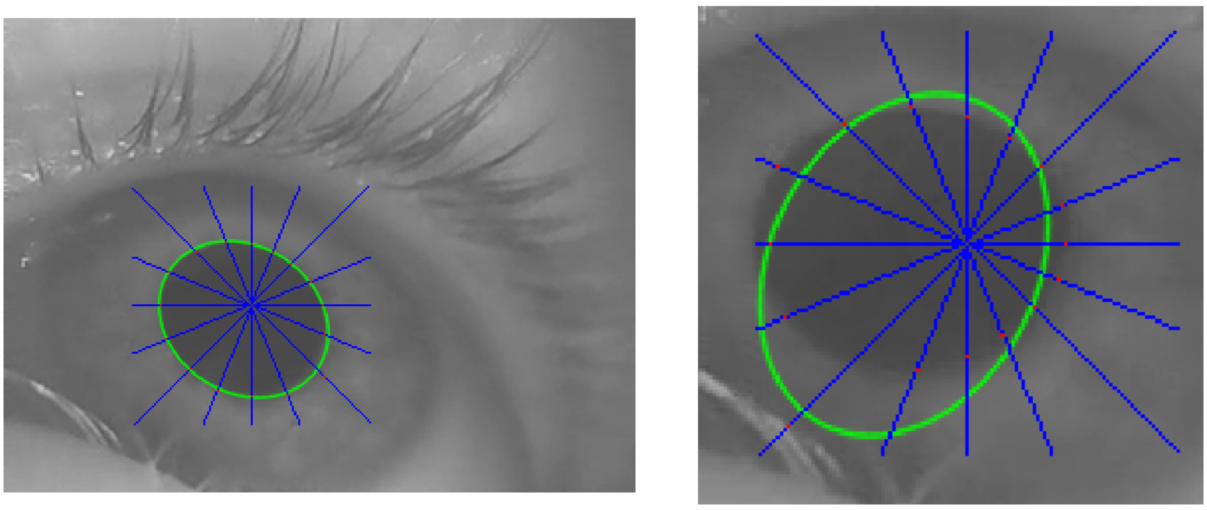
The algorithm was tested with pictures that were inserted at compile time with the parameters of the ellipse as output. The result was then compared with the result. This is not efficient and is addressed in the next section.

4.2. Improve Testing

It is very time consuming to generate picture data that can be compiled and to compare the output of the program with what is expected. A better way is when a tool can read a video file and display the result of the algorithm visually on each picture.

The program "Gazelle View" that was developed during the project study can already open and play video files. Because of the modular design of Gazelle View it is simple to insert the algorithm during the decode phase and paint the rays, the detected edges and the fitted ellipse on the image.

All data that was produced by the algorithm may also hold important information to find bugs or problems with the algorithm. In order to display that data besides the frame all data is packed into a struct and displayed in a Treeview beside the frame.


(a) Fitting of Ellipse without outliers

(b) Fitting of Ellipse with one outlier

Figure 4.1.: Comparison between results with and without an outlier. It shows the big impact of an outlier on the least square fitting.

The next step to improve testing would be to automate it and ideally condense the performance of the algorithm to parameters such as the mean and variance of the accuracy.

This should be done on a wide variety of test data, at best with high frame rate pictures that were captured with the eye tracking system. For each of those pictures an ellipse has to be fitted manually and stored as a golden reference. As this needs to be done for a lot of pictures the process of generating the reference needs to be simplified.

4.3. Outliers

The supplied algorithm does not differentiate between edges that are from the pupil or other edges from hair or the eyelid. The algorithm works fine when every edge is a right one as seen in figure 4.1a. But as a consequence to the least square method a single false detection severely alters the fitted ellipse as it happens in figure 4.1b.

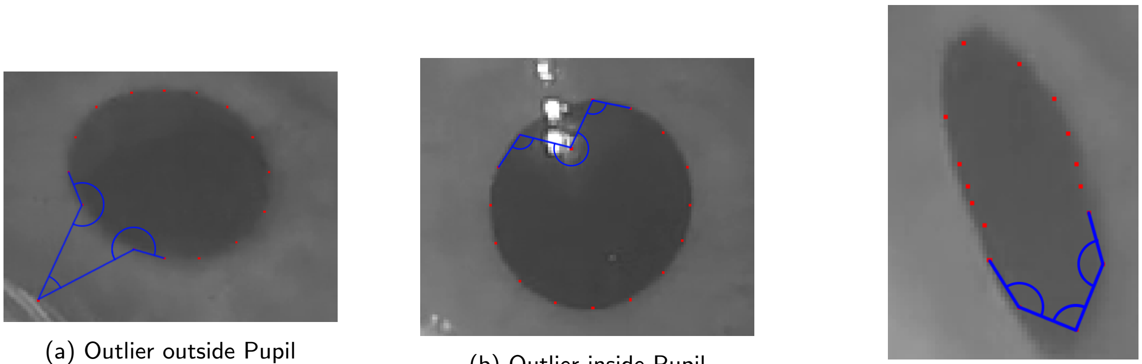
To mitigate the influence of outlier the occurrence needs to be reduced accompanied by the detection of outliers that are still detected.

4.4. Angle-Validation

To detect outliers it is necessary to find properties that differ strongly from the other transitions. One of those properties gets visible when the angle for each transition gets calculated with the neighbouring transitions.

In figure 4.2a a outlier lies on the bottom left corner of the image and outside of the pupil. The angle of the outlier is very small compared to angle of the neighbouring transitions. Those on the other hand have a greater angle than the other transition points.

When a outlier is inside the pupil then the situation is reversed as visible in figure 4.2b where the outlier has a big angle and the adjacent transitions very small angles.


(a) Outlier outside Pupil

(b) Outlier inside Pupil

(c) Accurate Transitions with small Angles

Figure 4.2.: Angles between Transitions around Outliers

Multiple adjacent outliers differ in that it is not possible to make a clear statement what the angle will be. But the neighbours to the Outliers still have the same properties as described in the last paragraphs. For that reason an algorithm that wants to detect outliers should focus on detecting the neighbours of outliers.

In figure 4.2c three adjacent transitions have angles that appear too small compared to its neighbours. None of them are outliers.

Overview of Algorithm

The algorithm to detect outliers based on the angles between transitions involves four steps:

1. Calculate angles
2. Find 3 adjacent angles with small differences
3. Categorize angles
    - To big
    - To small
    - As expected
4. Determine outliers

Calculate Angles

The angle between the points is calculated with the vectors that stretch from the center point to the other points. For each vector the angle between the x axis and the vector is calculated with the atan2 function. The advantage of the atan2 over the conventional arctangent function is that it takes coordinates and returns the angle for the correct quadrant. The difference of the resulting two angles is the desired angle between the points.

When the detected transitions are close to each other, measurement uncertainty can greatly affect the angles at such points. To mitigate that effect the angle for a point is only calculated with transitions that are distant enough.

Find adjacent angles with small differences

Context is important to categorize an angle. Only angles that are as expected do not always require context. This is the case when there are three adjacent angles that fall within a narrow range of each other and create context for the neighbouring angles.

Categorize Angles

The found context leads the way to iterate over every transition and categorize the angle within the context. This is done as described in the decision making diagram in 4.3. Angles are classified into three groups. Angles that are as expected, too big or too small.

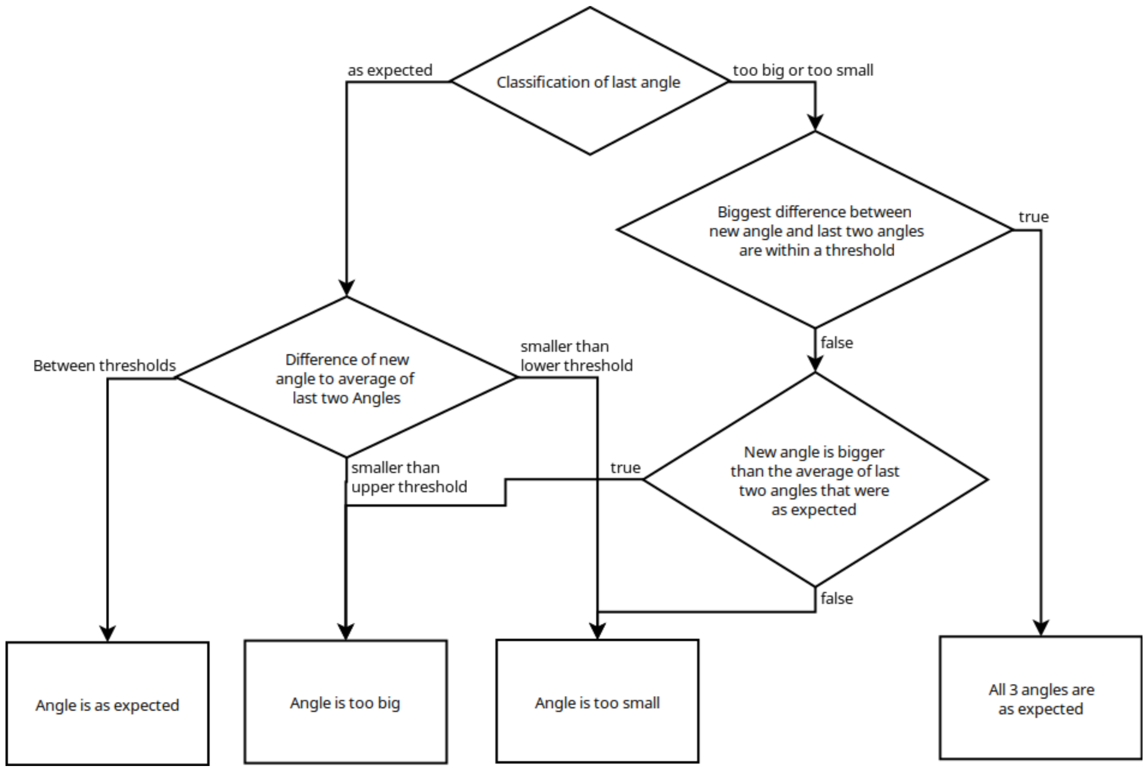

Classification of last angle
as expected
too big or too small
Biggest difference between new angle and last two angles are within a threshold
true
false
Difference of new angle to average of last two Angles
Between thresholds
smaller than lower threshold
smaller than upper threshold
New angle is bigger than the average of last two angles that were as expected
true
false
Angle is as expected
Angle is too big
Angle is too small
All 3 angles are as expected

Figure 4.3.: Decision making diagram for categorizing angles

Determine outliers

As described in section 4.4 outliers are surrounded by angles that are not as expected. The categorized angles make that determination straight forward. An outlier is a transition where neither the angle before or after is categorized as expected. The situation described in figure 4.2c where accurate transitions build three adjacent angles that are smaller than expected constitute an exception to this rule. This exception only occurs with adjacent small angles with neighbours that are as expected.

4.5. Colour of the Startpoint

The pupil has an uniform colour. So transitions that go from the pupil to the iris start with a colour close to the colour of the startpoint. Transitions from the iris to an eyelid on the other hand start with a colour that is brighter. This correlation can be exploited to improve the number of correctly detected transitions by dividing the score during the filter by the difference to the colour of the startpoint. To avoid a division with zero the absolute difference with an offset is used. Th offset can also be used to control the influence of this part of the algorithm.

4.6. Border Detection

The colours of te starburst are represented by the values between one and 255. When a point of the starburst is outside the image it is set to zero. With that it is simple to detect the border during the filter. The behaviour in this case is different depending if there was already a edge or not. If there was already an edge we would like to keep that edge otherwise the result of this ray should be discarded.

To decide if an edge has already ocured a similar method as in section 4.7 is used. But instead of adjusting the score of a transitions a hard threshold is applied.

4.7. Exploit good Results

The high framerate of the eye-capturing cameras results in small differences of the pupil position between frames. This can be used to improve the result because we already have an idea what the ellipse will look like. The startpoint can be adjusted to be the center ofthe old ellipse so the detected edges will be more equally distributed.

With the old ellipse it is also possible to calculate where a ray would transits this ellipse. This information is used to weight the score of a transition similar as the colour of the startpoint in section was handled. The score is divided by the absolute distance in pixel to the old ellipse on the ray with a small offset.

4.8. Fixed Threshold

This section is about a method that replaces the ray transition that uses a filter with a threshold.

An edge is detected when the difference to the colour of the startpoint is greater than a certain threshold. This approach is less flexible than the filter because different light conditions or eyes might require a different threshold.

5. Conclusion / Results

But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure? On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.

But I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure? On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue; and equal blame belongs to those who fail in their duty through weakness of will, which is the same as saying through shrinking from toil and pain. These cases are perfectly simple and easy to distinguish. In a free hour, when our power of choice is untrammelled and when nothing prevents our being able to do what we like best, every pleasure is to be welcomed and every pain avoided. But in certain circumstances and owing to the claims of duty or the obligations of business it will frequently occur that pleasures have to be repudiated and annoyances accepted. The wise man therefore always holds in these matters to this principle of selection: he rejects pleasures to secure other greater pleasures, or else he endures pains to avoid worse pains.

Declaration of primary authorship

I / We hereby confirm that I / we have written this thesis independently and without using other sources and resources than those specified in the bibliography. All text passages which were not written by me are marked as quotations and provided with the exact indication of its origin.

Place, Date:	[Biel/Burgdorf], 07.02.2014	
Last Name/s, First Name/s:	[Test Peter]	[Müster Rösä]
Signature/s:	....	....

Glossary

BibTeX Program for the creation of bibliographical references and directories in TEXor LATEXdocuments.

Index Index with keywords from text.

Bibliography

[1] B. A. Herbert Gross, Fritz Blechinger, *Handbook of Optical Systems, Survey of Optical Instruments*. Wiley, 2008.

List of Figures

List of Tables

APPENDICES

A. Guide to Generate an Animation

1. Organise a PC, laptop or server that is powerful enough to render and runs linux.
2. Install blender, git, ffmpeg and make on it.
   - On Ubuntu or Debian this is done with
     ```
     apt install blender make git ffmpeg
     ```
   - On Arch Linux this is done with
     ```
     pacman -S blender make git ffmpeg
     ```
3. Clone the git repository for where you want to work
   ```
    git clone https://github.com/timoll/eye-generator
   ```
   In case you want to contribute to the project, fork the project on github and clone your repository. Once you made meaningful changes you can push them to your repository and create a pull request.
4. Change directory into the cloned project
   ```
   cd eye-generator
   ```
5. There are already a few animations stored in the repository. Open a .json file for reference.
   ```
   gedit animation.json
   ```
   You can adjust the values manualy, write a script that generates the animation you want or modify it on `http://jsoneditoronline.org/`

Eye Position Left/Right	Position of the Left/Right Eye in centimeters.	
Diameter	Diameter of the eye in millimetres	
Last Frame	The last frame that will be rendered	
Right/Left Eye Keyframes	Keyframes of the Left/Right eye. Frames in between are interpolated	
	Frame	When the eye has this position
	Rotation	The direction the eye is looking {-90, 0, 0} is forward. Change x for up/down and z for left/right
Glasses Keyframes	Keyframes for the glasses	
	Frame	When the eye has this position
	Position	Position of the Glasses {0 , 0, 0} is a good start
	Rotation	The direction the glasses is pointing {-90, 0, 0} is forward.

6. Save the new json file in the same folder as the rest of the project

7. Export your file so make knows it

   ```
   export ANIMATION_JSON=myanimation.json
   ```

8. Create the different blend files with your animation

   ```
   make
   ```

9. (optional) Verify that your animation is right in blender

   ```
   blender leftup.blend
   ```

   In the bottom left corner select "Timeline" and play the animation. You can zoom out or in by scrolling and move around by pressing the middle mouse button.

10. Start the render, make sure this is done on the server if you have access to one.

    `make render` It will start the render detached so you can continue to use the command line. If you want to abort the render you need to stop blender

    ```
    pkill blender
    ```

    Note: this will kill every instance of blender so make sure you save open blend files first.

11. Wait, this process may take some time, especially if the pc is not that fast.

12. Once all frames are rendered you can inspect them in their folders. The blender output is saved as log in log/renderlu.log or similar for each camera.

    The position and rotation of the glasses and the rotation of eyes for each frame are logged also in the log folder

13. (optional) You might want to generate a movie files from the pictures. For the eye cameras just run the script `./genffmpeg`. It will generate video files in the folder videos.

14. Cleaning up. Once you have saved the generated data that you need. Clean up the folder by running

    ```
    make clean
    ```

    To delete all the blend files that are not needed and

    ```
    make clean_render
    ```

    To delete all files that where generated by the render.

B. Additional Appendix

B.1. Test 1

To an English person, it will seem like simplified English, as a skeptical Cambridge friend of mine told me what Occidental is. The European languages are members of the same family. Their separate existence is a myth. For science, music, sport, etc, Europe uses the same vocabulary. The languages only differ in their grammar, their pronunciation and their most common words. Everyone realizes why a new common language would be desirable: one could refuse to pay expensive translators. To achieve this, it would be necessary to have uniform grammar, pronunciation and more common words. If several languages coalesce, the grammar of the resulting language is more simple and regular than that of the individual languages. The new common language will be more simple and regular than the existing European languages.

B.1.1. Environment

It will be as simple as Occidental; in fact, it will be Occidental. To an English person, it will seem like simplified English, as a skeptical Cambridge friend of mine told me what Occidental is. The European languages are members of the same family. Their separate existence is a myth. For science, music, sport, etc, Europe uses the same vocabulary. The languages only differ in their grammar, their pronunciation and their most common words. Everyone realizes why a new common language would be desirable: one could refuse to pay expensive translators. To achieve this, it would be necessary to have uniform grammar, pronunciation and more common words.

C. Content of CD-ROM

Content of the enclosed CD-ROM, directory tree, etc.