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

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus scelerisque, leo sed iaculis ornare, mi leo semper urna, ac elementum libero est at risus. Donec eget aliquam urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc fermentum nunc sollicitudin leo porttitor volutpat. Duis ac enim lectus, quis malesuada lectus. Aenean vestibulum suscipit justo, in suscipit augue venenatis a. Donec interdum nibh ligula. Aliquam vitae dui a odio cursus interdum quis vitae mi. Phasellus ornare tortor fringilla velit accumsan quis tincidunt magna eleifend. Praesent nisl nibh, cursus in mattis ac, ultrices ac nulla. Nulla ante urna, aliquet eu tempus ut, feugiat id nisl. Nunc sit amet mauris vitae turpis scelerisque mattis et sed metus. Aliquam interdum congue odio, sed semper elit ullamcorper vitae. Morbi orci elit, feugiat vel hendrerit nec, sollicitudin non massa. Quisque lacus metus, vulputate id ullamcorper id, consequat eget orci .

Contents

1. Introduction

1.1. Organising Documents

Eye tracking is used to measure where someone is looking. It is used in a wide variety of applications such as marketing research, psychology, virtual reality and sports training.

To enable high speed eye tracking for sports the HuCE developed an eye tracking system called the Gazelle Eye Tracker that is fast, portable and built for outdoor usage. The

For analyzing the recorded footage a player that can put an overlay over the video playback.

1.2. Requirements

There a few key requirements that can be categorized in optional and must have. This is done in 1.1.

Requirement	**must**	**optional**
Play video	x	
Video has overlay	x	
Step frame for frame forward and backward	x	
Step overlay for overlay forward and backward	x	
Overlay and Frames are in sync	x	
Display data of eye-cameras		x
Play at various playspeeds		x
Play overlays		x

Table 1.1.: List of requirements

2. Gazelle View

Gazelle View is the software developed in this project.

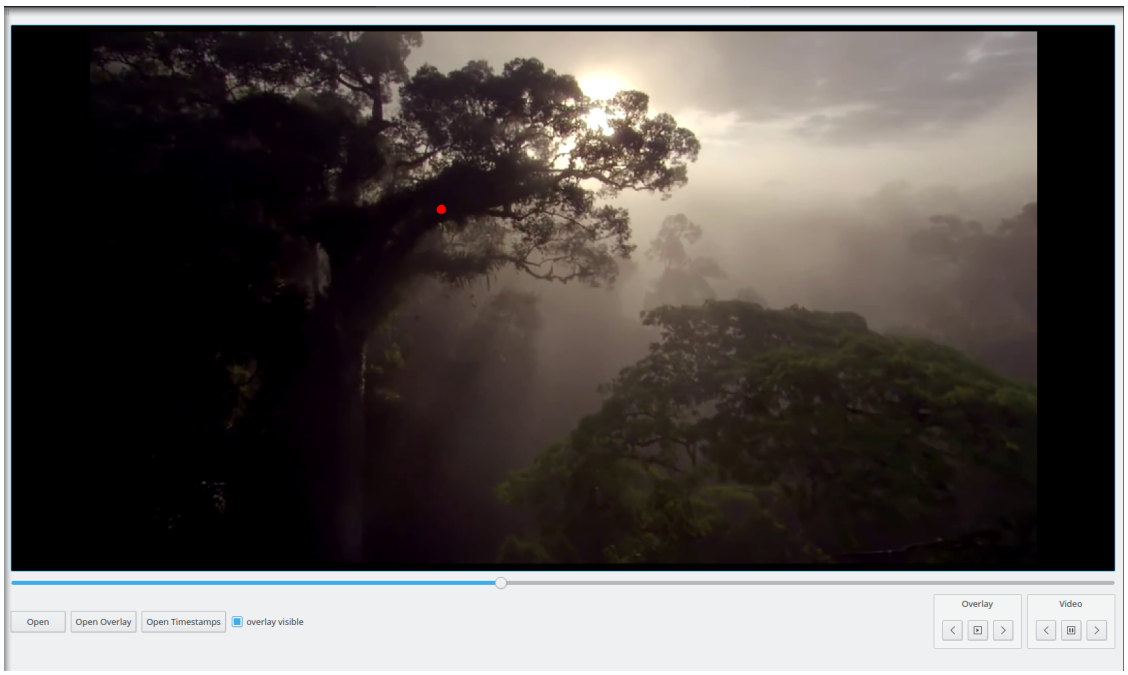

Figure 2.1.: GUI of Gazelle View with the Breeze icon theme

2.1. Use cases

The use cases for Gazelle View are presented in **??** and are deducted from the requirements.

- The user can open a video, overlays and timestamps
- The user can play the video
- The user can pause the video
- The user can go to the next or previous frame
- The user can play overlays
- The user can pause overlays
- The user can go to the next or previous overlay

2.2. Class Diagram

Gazelle View is build with four classes as seen in **??** The GUI is handled by the class MainWindow. User input is forwarded to the class VideoHandler. VideoHandler is in control of which overlay and frame are displayed or decoded and it also manages the buffer. Because decoding is an expensive operation, so VideoHandler orders frames to be decoded from DecodeWorker. DecodeWorker has as sole instance access to the videostream and has to provide metadata such as framerate and how total count of frames in the stream. DecodeWorker runs in it's own thread so it doesn't block the other classes. The class Overlays is responsible to parse overlays and timestamps and find the corresponding frame or overlay for a timestamp.

2.2.1. MainWindow

MainWindow handles the GUI, user input and displays the image with the overlay. It extends from QMainWindow and allows using the signal/slot framework from Qt. MainWindow sets up the icons of the play/pause, back and forwards buttons for overlay and video so that they use the icon theme of the current desktop environment. Fallback icons are provided for system that don't have themes or themes that don't have the necessary icons. An example of the GUI with the Breeze icon theme from KDE can be seen in 2.1.

The communication with the class VideoHandler is done over the signal/slot framework from qt. The slot "displayImage" gets called when VideoHandler want's to display a new image. This image is shown on a QGraphicsScene with black background.

2.2.2. VideoHandler

VideoHandler handles a few things.

- Order new frames to be decoded
- Send a frame to be displayed
- Handle the framebuffer
- Keep overlays and Frames in sync
- Send overlays to be displayed

VideoHandler extends QObject to communicate with the other classes over the signal/slot framework.

VideoHandler relies on DecodeWorker to decode the frames in a separate thread and on Overlays for getting the information which overlay needs to be displayed on which frame.

Framebuffer

As decided in 3.2.2 this is done with a QHash with the framenumber as key and a pointer to the Image as value. The buffer is trailing so it stores the last 50-100 frames. This is because the stream can only be decoded forward and jumping to a previous frame is expensive. To keep the size of the framebuffer reasonable a method iterates over the QHash when it get's to large and deletes all frames that are more than 50 frames behind.

2.2.3. Overlay

The class Overlays is responsible for the following things.

- parse file with information about overlays
- parse file of timestamps for the frames
- store overlays and timestamps
- return the overlay before or after a timestamp
- return the frame before a timestamp
- return the timestamp for a certain frame

The unit of the timestamps don't matter as long as they are the same for overlays and frames. As discussed in 3.2.2 a QMap stores the information for the overlays. The next or previous overlay is accessible for a timestamp. Additionally the corresponding timestamp for the overlay is also returned.

Timestamps for Frames

It is required to provide access for both the timestamp for a certain frame and the frame for a certain timestamp. In 3.2.2 successive approximation was proposed to solve this issue. The implementation is as follows:

```
for (int i = size/2; i >= 1; i /= 2) {
    if ((frameCount > (currentFrame + i)) &&
            (_sceneFrames.at(currentFrame + i) < timestamp)) {
        currentFrame += i;
    }
}
```

- `_sceneFrames` is the QVector that stores timestamps
- `frameCount` is the size of `_sceneFrames`
- `size` is the next higher power of two after `frameCount`
- `currentFrame` holds the result at the end of the algorithm
- `currentFrame + i` is the frameindex that is tested

The result converges to the frameindex that has the timestamp below the timestamp that is supplied. It does so by comparing the supplied timestamp with the timestamp at index $\frac{\texttt{size}}{2}$ of `_sceneFrames` first and followed by either $\frac{\texttt{size}}{4}$ or $\frac{3 \times \texttt{size}}{4}$ depending on the outcome of the comparison. This goes on until the increment is one.

2.2.4. DecodeWorker

The job of the DecodeWorker is to decode frames and convert them into a usable format so that the frame can be displayed on screen on a QGraphicScene. A library that enables decoding frames and give access to the pixeldata of each frame is OpenCV. It would also be possible to play videos with QMultiMedia. The disadvantage of this is that there is no direct access to each frame and the pixel data. OpenCV was chosen because i in the future it might be a requirement to manipulate or analyze each frame of the stream and OpenCV provides many tools that enable that.

Decoding and converting a video stream takes time so DecodeWorker is moved to it's own thread so it doesn't block the GUI or other parts of the program. Resulting race conditions are mitigated by the use of QMutex.

Getting a Frame Ready

The frame needs to be stored as a QPixMapItem in order to be displayed on a QGraphicsScene. This is done by a few steps:

1. Decode a single frame with OpenCV into a Mat
2. Convert the Mat from BGR to RGB
3. transfer the pictur data from the Mat to a QImage
4. convert the QImage to a QPixMap

2.3. 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 TeXnicCenter program. Under the menu "Build" > "Define Output Profile..." (short: alt + F7) in the "Postprocessor" register, the window shown in Figure 2.2 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"
```

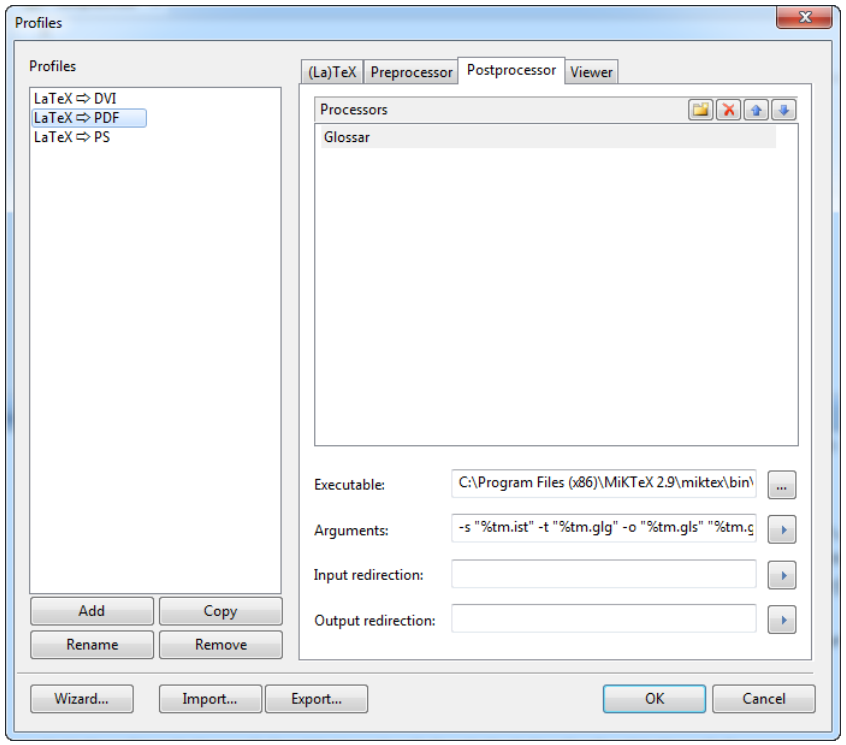

Figure 2.2.: Post-processing

2.4. 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. Software Design

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regelialia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar.

3.1. Software Architecture

The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didnâ€™t listen. She packed her seven versalia, put her initial into the belt and made herself on the way. When she reached the first hills of the Italic Mountains, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way. On her way she met a copy.

$$\mathcal{N}(x \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{(1/2)}} \exp\left(-\frac{1}{2}(x-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})\right) \tag{3.1}$$

The copy warned the Little Blind Text, that where it came from it would have been rewritten a thousand times and everything that was left from its origin would be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the copy said could convince her and so it didnâ€™t take long until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agency, where they abused her for their projects again and again. And if she hasnâ€™t been rewritten, then they are still using her.

3.2. Design Decisions

This is a typo dummy text. On it you can see if all the letters there are and how they look. Sometimes one uses words like Hamburgefonts, Rafgenduks or Handgloves to test fonts. Sometimes phrases that contain all letters of the alphabet - one calls these sets "pangrams".

Well known is this: The quick brown fox jumps over the lazy old dog. Often in type dummy texts also foreign-language sentence parts are installed (AVAIL™ and Wefox® are testing aussi la Kerning) to test the effect in other languages. In Latin, for example, almost every font looks good.

3.2.1. Storage Containers

Qt has an implementations for all popular storage containers. They all have different structures and are intended for different applications. The Containers can be categorized in two major categories. The first are sequential container each element is stored at a certain index and can be accessed over said index. The second are associative containers and set where all elements are stored with a key. The algorithmic complexity of the containers are shown in Table 3.1 for the sequential containers and in Table 3.2. O(n) stands for linear time so it scales linearly with the size of a container. O(1) stands for constant time, so each operation takes a fixed amount of time no matter how big the container is. O(log(n)) equals linear time. So the cost of an operation on a container raises logarithmic to the number of items in that container. If the behavior is not guarantied it is specified as amortized.

	Index lookup	Insertion	Prepending	Appending
QLinkedList<T>	O(n)	O(1)	O(1)	O(1)
QList<T>	O(1)	O(n)	Amort. O(1)	Amort. O(1)
QVector<T>	O(1)	O(n)	O(n)	Amort. O(1)

Table 3.1.: Algorithmic Complexity of the sequential Qt Containers where each element can be accessed over an index [1]

	Key lookup		Insertion	
	Average	Worst Case	Average	Worst case
QMap<Key, T>	O(log n)	O(log n)	O(log n)	O(log n)
QMultiMap<Key, T>	O(log n)	O(log n)	O(log n)	O(log n)
QHash<Key, T>	Amort. O(1)	O(n)	Amort. O(1)	O(n)
QSet<Key>	Amort. O(1)	O(n)	Amort. O(1)	O(n)

Table 3.2.: Algorithmic Complexity of the associative Qt Containers where each element can be accessed over the associated key [1]

QLinkedList

QLinkedList stores elements by pointing to the previous and next elements, creating a chain that is linked together. Because the pointer to an element is only stored in the adjacent elements, it is slow when accessing an element over an index. However when accessed over an iterator it is possible to insert an element at any point during the traversal of the linked list.

QList

QList allows for fast access over an index and allocates space before and after its internal array which usually results in constant time insertion on both ends of the list as shown in 3.1. It has to be noted that QList allocates its elements on the heap when they use more storage space than a pointer.[2]

QVector

QVector is similar to QList as it can also be accessed over an index in constant time. Because QVector doesn't allocate storage before the array it is not possible to prepend objects in constant time. QVector stores each element successively.

QHash

QHash uses a hash table to store its associated key-value pairs. The advantage of that is fast insertion and lookup of a key, usually in constant time. The hash table is not sorted, so an item with the key "15" can be followed by an item with the key "4". QHash automaticly grows and shrinks according to the number of stored elements.

QMap

QMap is similar in the usage as a QHash. The main difference is that QMap sorts the keys. So an item with the key "4" is followed by an item with the key "15" when there is no key between the two values. The trade-off for that is slower lookup and insertion of logarithmic time.

QMultiMap

QMultiMap has the same properties as a QMap but is more convenient if you want to store multiple values per key.

QSet

QSet is similar to QHash but it does not store a value. An application for a QSet is to store a stream of values without storing duplicates.

3.2.2. Containers in Gazelle View

In Gazelle View there are multiple sets of data that needs to be stored. There are the decoded Frames that need to be handled dynamicaly, the position an the timestamp of each overlay and the timestamp for each frame. Each of those datasets has different requitement and the best storage container needs to be evaluated.

Framebuffer

Each frame contains a large amount of Data. Around 6 MB per picture for a fullHD stream. Additionally it is also important to store the associated framenumber for each frame. As the buffer is usually a continuous flow of data this can be done with a sequential container with an offset or an associative container with the key represented as frame. Disadvantage of sequential containers are that if you move the first stored frame to the first index the container needs to be rearranged each time frames are deleted. If the frames are stored at the index of their framenumber you have a very large container with mostly empty elements and you need to keep track where how many elements are stored to free the buffer efficiently.

Because the frames don't need to be sorted it is best to use a QHash to store a frame with the framenumber as key. It allows for uncomplicated insertion and checks of how many frames are already stored for handling of the buffersize.

Overlays

An overlay consists of the position where it should be placed and the timestamp when it should be displayed. Access is usually over a timestamp to get an overlay that is before or after said timestamp. To accomplish that the container needs to be associative and sorted. QMap fulfills those requirements perfectly.

Timestamps

The timestamps consists out of a continuous list of timestamps for each frame. It needs to be possible to access the timestamp for each frame and the frame before a timestamp. There are a few options to solve this dual access problem.

The first is a QVector where each timestamp is stored at the index of the corresponding frame. The advantage of that approach is that it is not memory intensive and access of a timestamp of a frame is done in constant time. But o get the frame for a timestamp requires iterating over the QVector resulting in linear time access. As there may be multiple of 10'000 frames per video this is not desirable.

Another option would consist of a QVector for the timestamps and a QMap for accessing the framenumber from a timestamp. The access over a QMap reduced the time needed from linear to logarithmic time for the cost of duplicating the stored information.

A third option exploits the fact that timestamps count up. So it is possible to get the correct frame out of timestamps stored in a QVector in logarithmic time with the usage of successive approximation to calculate the frame. As this combines the small memory footprint as the first and fast access of the second option it is the best choice for this problem.

3.3. Webstandards

+

Everywhere the same old story. The layout is complete, the text is slow in coming. This layout is now not naked in space and small and empty occurs, I help out: the dummy text. Created exactly for this purpose, always in the shadow of my big brother "Lorem Ipsum", I look forward every time you read a few lines. Because esse est percipi - being is to be perceived.

And now because you already have the goodness to accompany me a few more sentences long, I would like to take this opportunity to serve you not only as a stopgap, but to point out something that is going to be perceived as deserved: Web viz. See Web standards are the rules that build on the websites. So there are rules for HTML, CSS, JavaScript or XML, words that you might have heard of your developers. These standards ensure that all parties the maximum benefit from a website.

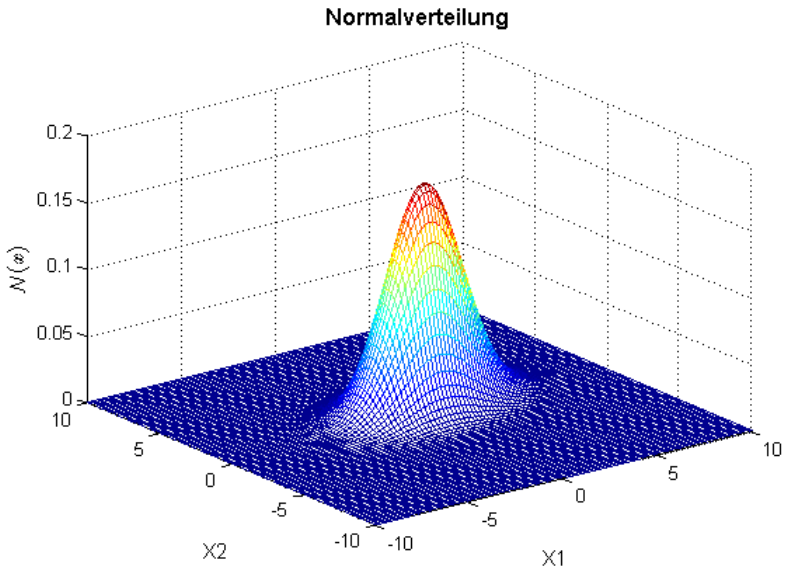

Figure 3.1.: Normal distribution

In contrast to previous websites we no longer need, for example, two different sites for the program Internet Explorer and another browser. It extends a page that - properly applied - both works on different browsers on the net, but just as good for printing or display on a cell phone is. Mind: A site for all formats. What a relief. Standards save time provide for the development costs and ensure that web pages can be easier to maintain later. Of course, only if everyone adheres to these standards.

4. Performance

Showing decoded pictures on the screen takes some time. The scene camera of the eye-tracking system record with 30 frames per second at a resolution of 1920x1080. Translated into time that means for each frame there are on average $\frac{1s}{30} = 33.3ms$ time for it to be decoded and converted so it is ready to be displayed.

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] T. Q. Company, "Qt documentation: Containers," accessed: 2016-05-16. [Online]. Available: http://doc.qt.io/qt-5/containers.html

[2] ——, "Qt documentation: Qlist," accessed: 2016-05-16. [Online]. Available: http://doc.qt.io/qt-5/qlist.html

[3] T. Erbsland and A. Nitsch, "Diplomarbeit mit LaTeX," 2008. [Online]. Available: http://drzoom.ch/project/dml/

[4] M. Jürgens, "Latex – Fortgeschrittene Anwendungen," 1995. [Online]. Available: http://www.fernuni-hagen.de/zmi/katalog/A027.shtml

[5] ——, "Latex – eine Einführung und ein bisschen mehr..." 2000. [Online]. Available: http://www.fernuni-hagen.de/zmi/katalog/A026.shtml

[6] "KOMAScript documentation project." [Online]. Available: http://koma-script.net.tf

[7] H. Kopka, *LaTeX, Band 1: Eine Einführung*. Pearson Studium, 2002, vol. 3.

[8] F. Marti, "Schreiben über Technik – Redaktion und Gestaltung von technischen Berichten und anderen technikbezogenen Texten," Berner Fachhochschule, 2006.

[9] B. Raichle, "Einführung in die BibTeX-Programmierung," 2002. [Online]. Available: http://www.dante.de/dante/events/dante2002/handouts/raichle-bibtexprog.pdf

[10] C. Schenk, "MiKTeX Project Page." [Online]. Available: http://www.miktex.org/

[11] S. Wiegand, "TeXnicCenter." [Online]. Available: http://www.TeXnicCenter.org/

List of Figures

List of Tables

Index

APPENDICES

A. Arbitrary Appendix

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. It will be as simple as Occidental; in fact, it will be Occidental.

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.