Algorithms for Eye Detection

Semester Thesis

Anton Jakob Paris

23. February 2023

Dr.-Ing. Martin Weisenhorn

Image Processing and Computer Vision , OST Rapperswil

Abstract

This example thesis briefly shows the main features of our thesis style, and how to use it for your purposes.

Contents

Chapter 1

Introduction

This is version `v1.4` of the template.

We assume that you found this template on our institute's website, so we do not repeat everything stated there. Consult the website again for pointers to further reading about LaTeX. This chapter only gives a brief overview of the files you are looking at.

1.1 Features

The rest of this document shows off a few features of the template files. Look at the source code to see which macros we used!

The template is divided into TeX files as follows:

1. `thesis.tex` is the main file.
2. `extrapackages.tex` holds extra package includes.
3. `layoutsetup.tex` defines the style used in this document.
4. `theoremsetup.tex` declares the theorem-like environments.
5. `macrosetup.tex` defines extra macros that you may find useful.
6. `introduction.tex` contains this text.
7. `sections.tex` is a quick demo of each sectioning level available.
8. `refs.bib` is an example bibliography file. You can use BibTeX to quote references. For example, read [1] if you can get a hold of it.

1.1.1 Extra package includes

The file `extrapackages.tex` lists some packages that usually come in handy. Simply have a look at the source code. We have added the following comments based on our experiences:

REC This package is recommended.

OPT This package is optional. It usually solves a specific problem in a clever way.

ADV This package is for the advanced user, but solves a problem frequent enough that we mention it. Consult the package's documentation.

As a small example, here is a reference to the Section *Features* typeset with the recommended *varioref* package:

> See Section 1.1 on the preceding page.

1.1.2 Layout setup

This defines the overall look of the document – for example, it changes the chapter and section heading appearance. We consider this a 'do not touch' area. Take a look at the excellent *Memoir* documentation before changing it.

In fact, take a look at the excellent *Memoir* documentation, full stop.

1.1.3 Theorem setup

This file defines a bunch of theorem-like environments.

Theorem 1.1 *An example theorem.*

Proof Proof text goes here. □

Note that the q.e.d. symbol moves to the correct place automatically if you end the proof with an `enumerate` or `displaymath`. You do not need to use `\qedhere` as with *amsthm*.

Theorem 1.2 (Some Famous Guy) *Another example theorem.*

Proof This proof

1. ends in an enumerate. □

Proposition 1.3 *Note that all theorem-like environments are by default numbered on the same counter.*

Proof This proof ends in a display like so:

$$f(x) = x^2.$$ □

1.1.4 Macro setup

For now the macro setup only shows how to define some basic macros, and how to use a neat feature of the *mathtools* package:

$$|a|, \quad \left|\frac{a}{b}\right|, \quad |\frac{a}{b}|.$$

Chapter 2

Data Set

2.1 Labelled Pupils in the Wild (LPW)

2.1.1 Description

The data set "Labelled Pupils in the Wild" [2] or short LPW was created by the Max Plank Institution and and contains 66 high-quality, high-speed eye region videos for the development and evaluation of pupil detection algorithms. All videos are labeled with the center of the pupil. 22 participant's eye region with five different ethnicities, five different eye colors were recorded.

The goal of the data set was to record samples of participants under conditions that are present in the reality. By having strong reflections, wearing glasses, wearing make up and so on the data set becomes a difficult challenge for pupil detection algorithms and a good evaluation of the algorithms is possible.

Figure 2.1: three example frames from the LPW data set.

2.1.2 Procedure

The participants were asked to look at a moving red ball as it moved around. The recording location was randomly picked and was in and around several buildings. Each location was chosen once, creating a diverse range of real life situations.

For the recording a high-speed Pupil Pro head-mounted eye tracker was used that took 95 frames per second with a resolution of 640x480 pixels. With this frame rate even fast eye movements last through several frames making it more robust to detect the pupil.

Location	Number of Videos
Outside	34.3%
Inside	65.7%

Table 2.1: Location of recordings

Light source	Percentage of recordings
Natural light	84.7%
Artificial light	33.6%

Table 2.2: Light Source of recordings

2.1.3 Ground truth annotation

In many cases the pupil area has a clear boundary an can be annotated easily. But in some difficult scenarios it was done manually. Because the participants followed a red fall with their eyes, there was additional information that was used to create the labels.

The complete data set has labels for the center of the pupil for every frame. Given the ground truth annotation, it is possible to evaluate algorithms and compare them to each other.

2.1.4 Folder structure

The folder stands for the part, file number stands the participants and the file extension for the file type. The text file contains the ground truth annotation of the pupil center in x and y coordinates for each frame.

The labels.ods file contains the information about the location, light source, vision aid used, prescription, nationality, eye color and gender. The README.txt file contains the information about the data set and the folder structure.

The folder structure is shown in the following tree diagram.

```
LPW/
├── labels.ods
├── README.txt
└── 1
    ├── 1.avi
    ├── 1.txt
    ├── 4.avi
    └── 4.txt
```

```
    ...
2
    4.avi
    4.txt
    ...
...
    ...
22
    ...
```

2.2 Eye characteristics

2.2.1 Anatomy of the eye

The eye is surrounded by the **eye lid**. Its purpose is to shield the eye from debris and lubricate the eye by spreading tears over its surface with each blink. Then there are four different parts of the eye that matters for pupil detection. The white part of the eye is the **sclera** and is separated from the **iris** by the **limbus**.

Name	Radius	Additional Information
Iris	12 mm	in average
Pupil	2-9 mm	varies with light intensity

Table 2.3: Oversight iris and pupil radius

The iris regulates the radius size of the **pupil** is and thereby regulates how much light comes through the pupil. The pupil is in the center of the iris and is the black part of the eye. The Darker the environment is, the larger the pupil radius. The Iris is colored and can be blue, green/hazle, brown, gray, amber and black. There are also other colors but they are in connection with health issues or genetic defects and play no role in this thesis.

The most important fact for pupil detection is that the iris, pupil and sclera have different brightness values. The pupil is the darkest area in the eye, the sclera is the brightest and the iris is in between. Depending on the color of the iris, the brightness value can vary. Those characteristics are often used in pupil detection algorithms and will be discussed in the next chapter. There can also be reflections of the environment be seen in the eye. This reflection is called **purkinje reflection** and can temper with pupil detection algorithms accuracy and therefore is a challenge for pupil detection algorithms.

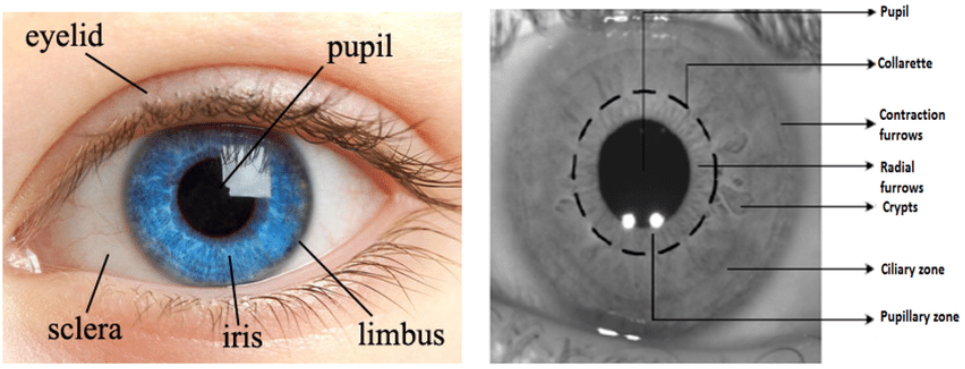


Figure 2.2: Overview of the different parts of the eye.

2.2.2 Image characteristics

Histogram

When inspecting the histogram of a randomly chosen frame from the LPW data set, it can be seen that the histogram shows different peaks in different intensities. In this particular example it can be seen that the peak between 0 and 50 corresponds to the pupil it self. This can be be seen in figure 2.3 because when comparing both histograms the lowest peak stays unchanged. The intensity of the iris is therefore between 90 and 150.

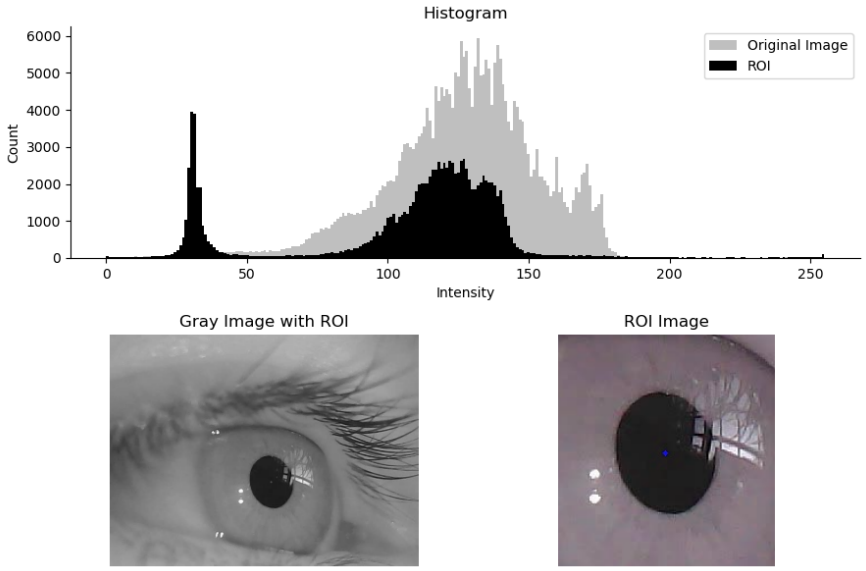


Figure 2.3: Comparison histogram of the whole image and the region of interest.

The sclera is the brightest part of the eye but in this frame the skin intensity will be around the same magnitude as the sclera intensity. Subtracting

the ROI from the original picture will result in a histogram almost only containing the skin and sclera. The histogram is therefore a strong tool for eye segmentation.

If there is more reflection on the pupil, the histogram will also change in its shape. The peak created by the pupil will shrink and flatten out. The mean of the histogram will increase as dark points are substituted by brighter points. This ultimately makes it hard for using a fixed threshold for segmentation. Using Histogram equalization increases the contrast of the image but stretches the peak into a wider intensity range.

Also interesting to inspect is one pixel row of the image with their intensity values. Here it can be observed that the pupil creates a valley in the intensity values. The reflection on the pupil generates a peak right after the valley. This can be seen in figure 2.4.

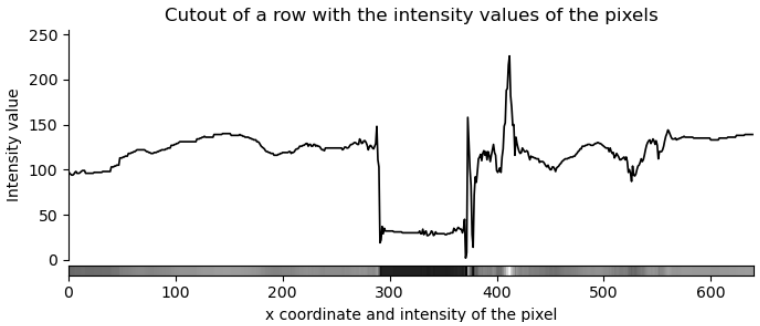


Figure 2.4: Plot of the intensity values of one pixel row.

The x coordinate represents the pixel at this coordination and the y coordinate represents the intensity value of the pixel.

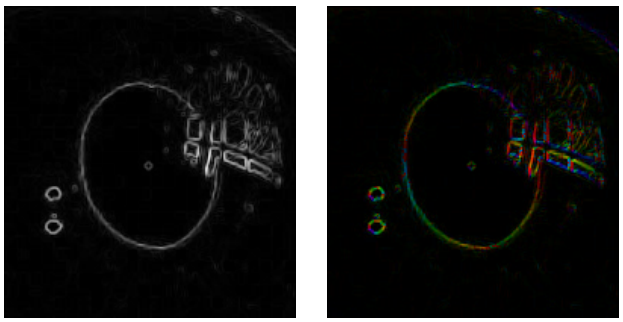
The conclusion is, that the histogram is a strong tool to gain insight information of the frame but can vary a lot depending on the environment, therefore adaptive algorithms have to be used.

Gradients

As already shown in the previous subsection, the pupil creates a valley in the intensity values. This is also reflected in the gradient of the image. The gradient responds strongly to the change of intensity in the transition from the pupil to the iris. The orientation of the gradient points from the pupil towards the iris and can be of great help for edge detection. The gradient is calculated by using the Sobel operator. The Sobel operator is a discrete differentiation operator. It convolves the image with a differential kernel and therefore emphasizes regions with high spatial frequency. The Sobel

operator will be discussed in a later chapter in more detail. But for know it's important to get an overview of the different characteristics of the eye.

(a) Image of ROI **(b)** Gradient in xG_x **(c)** Gradient in y G_y



(d) Magnitude G **(e)** Orientation θ

Figure 2.5: Plot of gradient characteristics of the ROI.

The gradient in x direction is shown in figure 2.5b and the gradient in y direction is shown in figure 2.5c. The magnitude of the gradient is shown in figure 2.5d and the orientation of the gradient is shown in figure 2.5e. Depending on the environment, the gradient itself could already sufficient for pupil detection. In image processing the gradient is often used for edge detection and all sorts of different filters. Used correctly it can be very powerful. The magnitude is calulated by the following formula:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2.1}$$

Because we have the gradient in x and y direction, we can calculate the orientation of the magnitude by using the following formula:

$$\theta = \arctan \frac{G_y}{G_x} \tag{2.2}$$

The orientation is calculated in radians and can be converted to degrees by multiplying it with $\frac{180}{\pi}$. The orientation is perpenticular to the edge and

points in the direction of the gradient. Because the pupil is darker than the iris, the gradient points from the pupil to the iris.

Noise

In the data set there are different kind of noise. The most common noise is the noise created by the camera. This noise is called **Gaussian noise**. Gaussian noise is a statistical noise having a probability density function equal to that of the normal distribution, which is also known as the Gaussian distribution. The noise is created by the camera and is therefore not dependent on the environment. The noise is therefore not a problem for the pupil detection algorithm.

Considering that the goal is to detect the pupil there can also be other noise that can be problematic. **Reflection** of the environment can have an impact on the accuracy of the algorithm and destroy information about the pupil. Also the **eyelashes** can be problematic. They are very dark and therefore temper with the edges of the pupil and iris. Especially when the eyelashes are in front of the pupil. Also if make up is used.

Another Noise term can be that the eye is not constantly open. The **eyelid** can cover parts of the pupil or even the whole pupil. Then also glasses or lenses can add noise to the image.

Chapter 3

Theory

3.1 Algorithms

This chapter will take a look at commonly used algorithms in image processing for edge detection, identifying areas of interest and applying them onto pupil detection. At first the algorithms will be discussed and analyzed on possible use cases, individual strengths and weaknesses. To show the nature of the algorithm, the same preprocessed images from the LPW data set are used and therefore it is possible to showcase the results and compare them.

3.1.1 Fundamental notation

Thru out this thesis the following notation is used to descripe the algorithms. The image with intensity level I is a function

$f(x,y) : \mathbb{N}^2 \to \mathbb{N}$, where $f(x,y)$ is the intensity $I \in [0,255]$ at position (x,y)

In image processing there the coordinate system is defined different than in mathematics. The origin is in the upper left corner and the x-axis is pointing to down vertically. The y axis is pointing to the right horizontally. this is shown in figure 3.1.

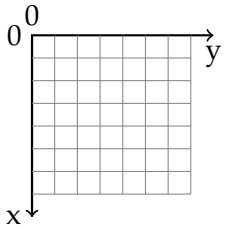


Figure 3.1: Coordinate system used in image processing.

Also important to note it that the image is a discrete function, therefore each intensity value I comes with an quantization error. This is also the case when using an algorithm on the intensity values of the image. So it is not possible to have an exact result, it is always an approximation of the real result.

Relationship between pixels

One also important theory in this thesis will be based on relationship between pixels. In this subsection the terms **neighborhood**, **adjacency**, **connectivity**, **region** and **boundaries** will be introduced and visualized, so that they can be used in the following chapters.

Neighborhood A pixel P at location (x, y) has two vertical neighbor pixels and two horizontal neighbor pixels in a 2D image. These neighbors are defined as $N_4(P)$ with coordinates:

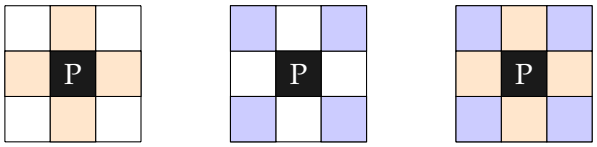
$$N_4(P) = \{(x, y+1), (x, y-1), (x+1, y), (x-1, y)\} \tag{3.1}$$

A pixel P at location (x, y) has four diagonal neighbor pixels in a 2D image. These neighbors are defined as $N_D(P)$ with coordinates:

$$N_D(P) = \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\} \tag{3.2}$$

Adding the neighbors from $N_4(P)$ and $N_D(P)$ results in the 8-neighborhood $N_8(P)$ of pixel P with coordinates:

$$N_8(P) = N_4(P) \cup N_D(P) \tag{3.3}$$



(a) $N_4(P)$ **(b)** $N_D(P)$ **(c)** $N_8(P)$

Figure 3.2: 3 different neighborhoods of pixel P at location (x, y).

Adjacency Mainly there are three types of adjacent pixels in a 2D image. Let V be the set of intensity values used to define adjacency. Depending on the intensity range of the image, it's possible to define different subsets of V, containing the intensity values that are considered as adjacent in the neighborhood. In a binary image V is often defined as $V = \{1\}$, where 0 stands for background and 1 for foreground (This can also be considered a binary mask). In a grayscale image V can be defined as any subset of the intensity range. In this thesis the intensity range is $V = \{0, 1, 2, ..., 255\}$.

To keep it simple, the following explanations will be based on a binary image with $V = \{1\}$. Let define P as a pixel at location (x, y) and Q as a pixel at location (x', y'). P and Q are considered adjacent if Q is in the neighborhood of P and $f(Q)$ is in V. These are the three adjacency types:

- 4-adjacency, if $Q \in N_4(P) and f(Q) \in V$: The pixels that are directly above, below, left and right of the pixel.
- 8-adjacency, if $Q \in N_8(P) and f(Q) \in V$: The pixels that are directly above, below, left, right and the pixels that are diagonally adjacent to the pixel.
- m-adjacency $\begin{cases} \text{if } Q \in N_4(P) \text{ and } f(Q) \in V \\ Q \in N_D(P) \text{ and } N_D(P) \cap N_4(Q) \text{ has no intensities} \in V \end{cases}$

(a) 4-adjacency **(b)** $8 - adjacency$

Figure 3.3: 3 different neighborhoods of pixel P at location (x, y).

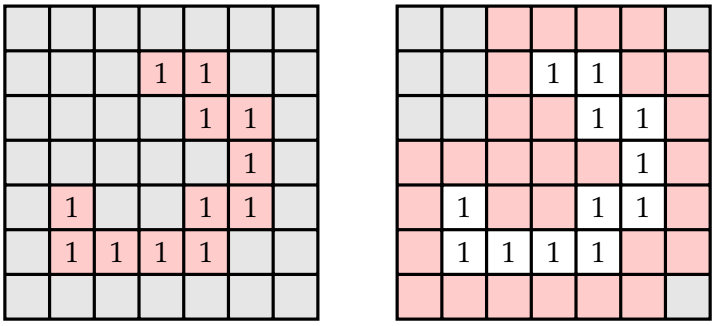
Connectivity The connectivity of point P is a set of points that can be reached in n steps with a given adjacency type and intensity set V. If point P is connected with Q then there exist a path (or curve) from P to Q that consists of a sequence of distinct pixels with coordinates

$$(x_0, y_0), (x_1, y_1), ..., (x_n, y_n), \text{ where } n \text{ is the lenght of the path.}$$

If the path is closed then $(x_0, y_0) = (x_n, y_n)$

Region Let's define R as an subset of pixels in an image. R is a region if all points $\in R$ are a connected set, meaning that all points in R are connected with each other and therefore form a region. This does not mean that the path connecting all points is closed. Two regions can be adjacent to each other, if their union again forms a connected set.

Boundary The outer boundary of a region R is the set of pixels not in R that are adjacent to pixels in R. In the definition of a boundary, the adjacency type is important. As a rule of thumb to define the boundary, the 8-adjacency is used. One important property of the outer boundary is, that it is a closed path. The inner boundary is the set of pixels that are in R but are adjacent to at least one pixel that is not in R and again the 8-adjacency is used. The inner boundary is not a closed path.



(a) region R **(b)** outer boundary (8-adjacent)

Figure 3.4: Region and outer border.

3.1.2 Preprocessing

To be able to compare the different approaches, it is important to define the used algorithms that lead to the wanted result. An image taken has to be preprocessed first. The idea behind this step is to create a common ground for narrowing the deviation of the images down, so that the algorithms are able to recreate the same result over span of different images.

Converting to grayscale

As already presented in the previous chapter, the only colorful part of the eye is the iris. But the color itself is of no interest for the detection. Therefore the frames are first converted to grayscale. This is done by converting the colors into a gray intensity value. During this chapter the grayscale images have certain properties:

Scaling	Shape	numpy array type
100%	640x480	unit8
50%	320x240	unit8
25%	160x120	unit8
12.5%	80x60	unit8
6.25%	40x30	unit8

Table 3.1: Scaling of the frames used in this thesis.

It is important to note that these resolutions are resolutions are congruent with the resolutions used in the LPW paper [2]. This is important for the comparison of the results. When scaling an image there is always an image interpolation done to find the best approximation for the new intensity value I. The interpolation used in this thesis is the bilinear interpolation.[**?**] This is a linear interpolation in the x and y axis of the intensity values and

solves equation 3.4. let $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{21} = (x_2, y_1)$ and $Q_{22}(x_2, y_2)$ be the four surrounding points. The intensity value I at (x, y) is then calculated by equation 3.4. The point P at (x, y) is the point of interest.

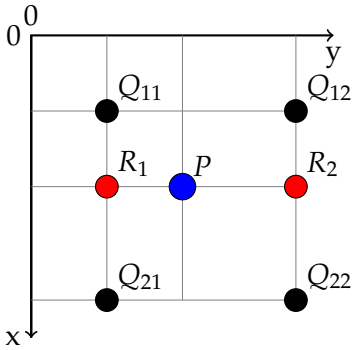


Figure 3.5: Bilinear interpolation.

$$v(x, y) = ax + by + cxy + d \tag{3.4}$$

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) = R_1(x, y_1) \tag{3.5}$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) = R_2(x, y_2) \tag{3.6}$$

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) = v(x, y) \tag{3.7}$$

Histogram equalisation

Another important aspect of preprocessing the frames is using Histogram equalization. This has the effect of increasing the contrast of the image. For this task Contras Limited Adaptive Histogram Equalization (CLAHE)[3] is used. CLAHE is an Histogram Equalization method that has the benefit that it is adaptive to the local contrast of the image. This is very useful if the contrast of the image is not uniform. By using Histogram Equalization on the frames, noise is added to the image. This can be dealt with by using a low pass filter, like an Gaussian filter for example.

Using a normal Histogram Equalization would lead to a loss of information in the region around the pupil and the iris. This is due to the fact that the contrast in this region is already very high. The CLAHE method splits the image into smaller blocks called ”tiles” and calculates the histogram on each block individualy and therefore does not lead to the same lose of important information at the pupil region. The CLAHE method is applied to the frames after they are converted to grayscale. The result of this step is shown in figure

3.6.

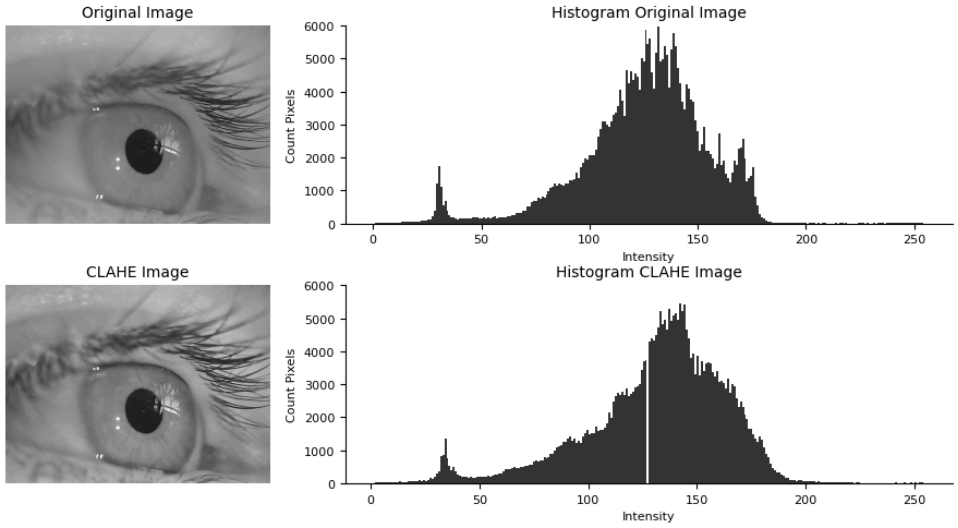


Figure 3.6: Example of CLAHE an its effect on the histogram.

These are the parameters used for the CLAHE plots

```
clahe = cv2.createCLAHE(clipLimit=1.0, tileGridSize=(11,11))
```

3.1.3 Edge Detection

The main goal behind edge detection is to find edges in the image. An edge is defined as a region that has a high contrast to its surrounding pixels, in other words a rapidly changing intensity in a small area. The edge detection is useful to filter the image for possible pupil contours. The edge detection analysis the image based on the change of intensity. Therefore a gradient calculation is used. There are different methods to calculate the gradient of an image. One of the most popular methods is the Sobel operator that makes use of the first differential of the image. The laplacian operator is another method that uses the second differential of the image. In this thesis the Sobel operator is used to calculate the gradient of the image. After calculating the gradient Canny edge detection is used to refine the edges. Canny edge detection is a multi step algorithm that uses a hysteresis thresholding to filter the edges. The result of the edge detection is a on pixel thick binary edge map. All edges are now possible candidates for the pupil contour.

Sobel Operators

The Sobel Operators are used is a common algorithm for edge detection. It is a gradient calculation that uses a 3x3 differential kernel to calculate the

gradient of the image. The Sobel gradient is calculated in x and y direction. The gradient in x and y direction are calculated by convolving the image with two different kernels. The image with gray values is defined as $f(x,y)$ and the kernels are defined as k_x and k_y. Therefore the gradient is calculated as follows:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{3.8}$$

$$k_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{3.9}$$

$$k_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \tag{3.10}$$

$$G_x(x,y) = f(x,y) * k_x = \sum_{s=-a}^{a} \sum_{t=-b}^{b} k_x(s,t) f(x+s, y+t) \tag{3.11}$$

$$G_y(x,y) = f(x,y) * k_y = \sum_{s=-a}^{a} \sum_{t=-b}^{b} k_y(s,t) f(x+s, y+t) \tag{3.12}$$

$G_x(x,y)$ and $G_y(x,y)$ are the gradients in x and y direction. $f(x,y)$ is the image and k_x and k_y are the kernels. The kernels convolved with the image $f(x,y)$ and the result is the gradient magnitude in x and y direction. In other words the convolution of an image with k_x or k_y gives as result the change from pixel to pixel in x or y direction. In python the Sobel in x and y are calculated with the OpenCV Library for example:

```
G_x = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=3)
G_y = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=3)
```

The total gradient magnitude G is calculated with this equation:

$$G = \sqrt{G_x^2 + G_y^2} \tag{3.13}$$

and the direction θ of the gradient is calculated with this equation:

$$\theta = \arctan \frac{G_y}{G_x} \tag{3.14}$$

The result of the convolution can be seen in figure 2.5 in chapter two.

Canny Edge Detection

The Canny Edge Detection is used to recieve single edge points form the gradient magnitude image. This algorithm can summarized in four steps[**?**]:

1. Noise reduction, smoothing the image with a Gaussian filter
2. Compute the gradient magnitude and direction
3. Non-maximum suppression to the gradient magnitude image
4. Use double thresholding and connectivity analysis to detect and link edges

Step 1: Noise reduction
The first step is to reduce noise from the input image. This is done by convolving the image with a low pass filter. For this task a Gaussian filter is used. The Gaussian filter is defined as:

$$f_{filter}(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.15}$$

The Gaussian filter is then convolved with the image so smooth the image.

$$f_{smoothed}(x,y) = f(x,y) * f_{filter}(x,y) \tag{3.16}$$

Step 2: Compute the gradient magnitude and direction
The gradient magnitude is calculated with the equation 3.13 and the gradient direction is calculated with the equation 3.14.

Step 3: Non-maximum suppression
The non-maximum suppression is used to thin the edges out, so that the edges are only one pixel wide. This can be achieved with an loop that goes over all edges and checks if the current pixel, belonging to the edge, is the local maximum in the direction of the $\pm$gradient vector. If that is the case the pixel is kept, otherwise it is set to zero. Because as already described in 3.1.1 the coordinate system is defined different. The gradient direction is in reference to the x axis.

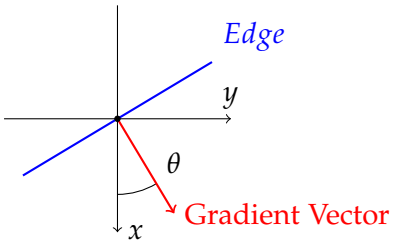


Figure 3.7: Definition of the gradient direction

Because an image is quantized, this also means that θ needs to be quantized to four directions to evaluate their neighbors.

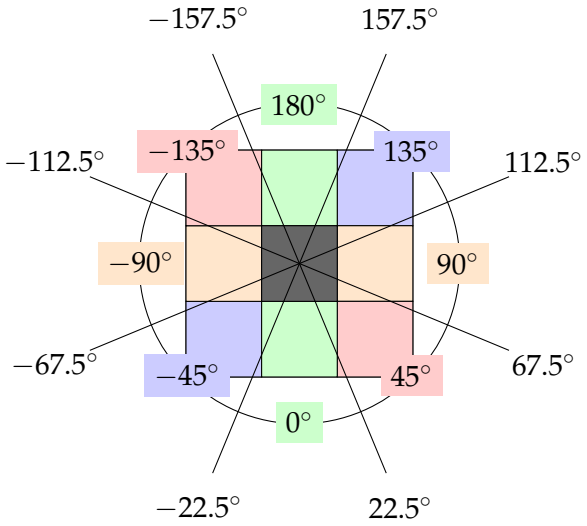
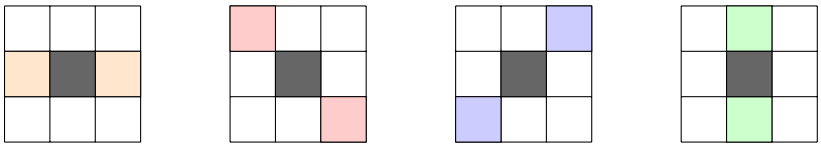


Figure 3.8: Quantization of the gradient direction

This leads following quantizations:

$$\theta_q = \begin{cases} 90, & \text{if } 67.5° < \theta \leq 112.5° \vee -112.5° < \theta \leq -67.5° \\ -45°, & \text{if } 22.5° < \theta \leq 67.5° \vee -157.5° < \theta \leq -112.5° \\ +45°, & \text{if } 112.5° < \theta \leq 157.5° \vee -67.5° < \theta \leq -22.5° \\ 0°, & \text{if } -22.5° < \theta \leq 22.5° \vee -157.5° < \theta \leq 157.5° \end{cases} \tag{3.17}$$

It is important to note that when following the gradient direction, the two neighboring pixels are used to evaulate the gradient magnitute maximum. This is shown in figure 3.8 and 3.9. If the gradient is maximal at the current pixel at (x, y), meaning it is a local maximum in the previous defined neighborhood in respect to the gradient direction, the value of the pixel is written into $g_n(x, y)$, otherwise it is set to zero $g_n(x, y) = 0$. This is called non-maximum suppression. Therefore $g_n(x, y)$ contaians only the thinned edges.



(a) Neighbors: 90° **(b)** Neighbors: 45° **(c)** Neighbors: -45° **(d)** Neighbors: 0°

Figure 3.9: The gradient magnitude is evaluated in the direction of the gradient.

Step 4: Double thresholding and connectivity analysis
After Step 3, $g_n(x,y)$ still shows edges that can be thicker than one pixel. $g_n(x,y)$ is then thresholded with a high an low threshold (hysteresis thresholding) creating two images:

$$g_{low}(x,y) = \begin{cases} g_n(x,y), & \text{if } g_n(x,y) \geq T_{low} \\ 0, & \text{otherwise} \end{cases} \tag{3.18}$$

$$g_{high}(x,y) = \begin{cases} g_n(x,y), & \text{if } g_n(x,y) \geq T_{high} \\ 0, & \text{otherwise} \end{cases} \tag{3.19}$$

Because two different thresholds were used, there is still overlap between g_{low} and g_{high}. All non zero pixels in g_{high} are considered strong edge pixels. To recieve all weak edge pixels, the strong edge pixels are substracted from g_{low}. The remaining pixels are considered weak edge pixels.

$$g_{weak}(x,y) = g_{low}(x,y) - g_{high}(x,y) \tag{3.20}$$

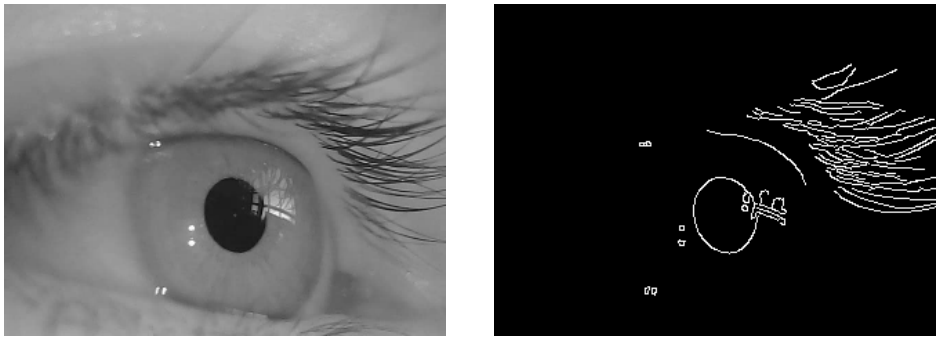
Next step is to connect the weak edge pixels to the strong edge pixels. This is done by checking the 8-neighborhood of each strong edge pixel. If there is a weak edge pixel in the neighborhood, it is considered a strong edge pixel. This is done until no more weak edge pixels are found. The result is a binary image $g_{final}(x,y)$ containing all edges. But this still does not return a one pixel thick edge.

To solve this the edges are passed on an edge-thinning algorithmus. Let's define the edges as set A and B as a structuring element. The equation for thining then becomes:

$$A \otimes B = A - (A \circledast B) \tag{3.21}$$

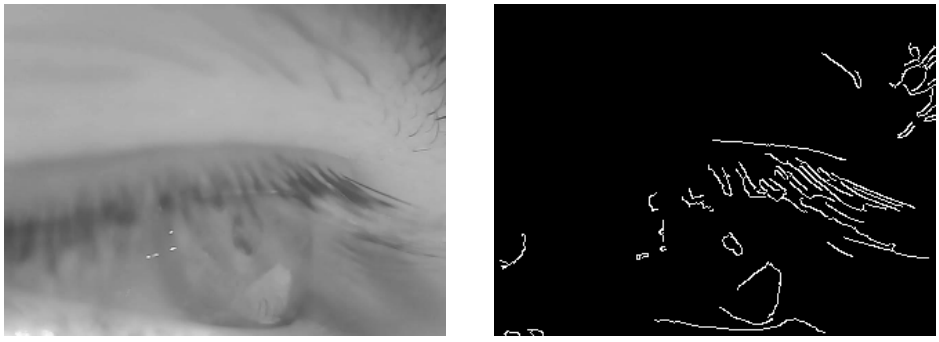
Where $\otimes$ is the thinning operator, $\circledast$ is the dilation operator.

Results In a frame where the pupil region clearly can be distinceted from the rest, the Canny edge detector can be used to find the pupil region. but as soon as more noise to the pupil is added, the hysteresis thresholding becomes more tricky and the detection accuracy decreases immensely. Also is it not possible to differentiate between the eye leashes, eye brows and the pupil. Therefore by using only the Canny edge detector. The pupil edges can not be found reliable and the algorithm itself is not adaptable to a great varriety of environments.



(a) Original image, scaling 0.5 **(b)** Clear pupil region, Canny edge detection

Figure 3.10: Canny edge detection on a clear pupil region



(a) Original image, scaling 0.5 **(b)** Weak pupil region, Canny edge detection

Figure 3.11: Canny Edge detection on a frame with weak pupil region

Chapter 4

Theory

In this chapter will take a look at commonly used algorithms in image processing for edge detection, identifyig areas of interest and refining them. We discuss those algorithms first in theory and then show it's use case in detecting the iris of an human eye.

To show the nature of the algorithm, the same preprocessed image is used and therefore it is possible to showcase the results and compare the algorithms in a later chapter

4.1 Algorithms

4.1.1 Preprocessing

To be able to compare the different approaches, it is important to define the used algorithms that lead to the wanted result. An image taken has to be preprocessed first. The idea behind this step is to create a common ground for narrowing the deviation of the images down, so that the algorithms are able to recreate the same result over span of different images.

Histogram matching

The algorithms are trimmed to a specific Histogram to work the best. Therefore all Images have to be preprocessed. To fullfil this requirement, Histogram matching is done. In the first step we take a look at the histogram used to match the others onto.

4.2 Gaussian blur

The idea behind using a Gaussian blur on a Image first is to minimize unnecessary information in to be processed image. By using the gaussian

blur the image loses details and this increases the chance to find the main edges in the image, finding the outline of the iris.

Dummy text.

4.2.1 Definition RANSAC

Dummy text.

4.3 Canny Edge Detection

Dummy text.

4.3.1 Definition Canny Edge Detection

Dummy text.

4.4 Active Contur

Dummy text.

4.4.1 Definition Active Countur

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Chapter 5

Writing scientific texts in English

This chapter was originally a separate document written by Reto Spöhel. It is reprinted here so that the template can serve as a quick guide to thesis writing, and to provide some more example material to give you a feeling for good typesetting.

5.1 Basic writing rules

The following rules need little further explanation; they are best understood by looking at the example in the booklet by Knuth et al., §2–§3.

Rule 5.1 Write texts, not chains of formulas.

More specifically, write full sentences that are logically interconnected by phrases like 'Therefore', 'However', 'On the other hand', etc. where appropriate.

Rule 5.2 Displayed formulas should be embedded in your text and punctuated with it.

In other words, your writing should not be divided into 'text parts' and 'formula parts'; instead the formulas should be tied together by your prose such that there is a natural flow to your writing.

5.2 Being nice to the reader

Try to write your text in such a way that a reader enjoys reading it. That's of course a lofty goal, but nevertheless one you should aspire to!

Rule 5.3 Be nice to the reader.

Give some intuition or easy example for definitions and theorems which might be hard to digest. Remind the reader of notations you introduced

many pages ago – chances are he has forgotten them. Illustrate your writing with diagrams and pictures where this helps the reader. Etc.

Rule 5.4 Organize your writing.

Think carefully about how you subdivide your thesis into chapters, sections, and possibly subsections. Give overviews at the beginning of your thesis and of each chapter, so the reader knows what to expect. In proofs, outline the main ideas before going into technical details. Give the reader the opportunity to 'catch up with you' by summing up your findings periodically.

Useful phrases: 'So far we have shown that . . . ', 'It remains to show that . . . ', 'Recall that we want to prove inequality (7), as this will allow us to deduce that . . . ', 'Thus we can conclude that . . . . Next, we would like to find out whether . . . ', etc.

Rule 5.5 Don't say the same thing twice without telling the reader that you are saying it twice.

Repetition of key ideas is important and helpful. However, if you present the same idea, definition or observation twice (in the same or different words) without telling the reader, he will be looking for something new where there is nothing new.

Useful phrases: 'Recall that [we have seen in Chapter 5 that] . . . ', 'As argued before / in the proof of Lemma 3, . . . ', 'As mentioned in the introduction, . . . ', 'In other words, . . . ', etc.

Rule 5.6 Don't make statements that you will justify later without telling the reader that you will justify them later.

This rule also applies when the justification is coming right in the next sentence! The reasoning should be clear: if you violate it, the reader will lose valuable time trying to figure out on his own what you were going to explain to him anyway.

Useful phrases: 'Next we argue that . . . ', 'As we shall see, . . . ', 'We will see in the next section that . . . , etc.

5.3 A few important grammar rules

Rule 5.7 There is (almost) *never* a comma before 'that'.

It's really that simple. Examples:

> We assume that . . .
> *Wir nehmen an, dass . . .*

It follows that ...
Daraus folgt, dass ...

'thrice' is a word that is seldom used.
'thrice' ist ein Wort, das selten verwendet wird.

Exceptions to this rule are rare and usually pretty obvious. For example, you may end up with a comma before 'that' because 'i.e.' is spelled out as 'that is':

For $p(n) = \log n/n$ we have ... However, if we choose p a little bit higher, that is $p(n) = (1+\varepsilon)\log n/n$ for some $\varepsilon > 0$, we obtain that...

Or you may get a comma before 'that' because there is some additional information inserted in the middle of your sentence:

Thus we found a number, namely n_0, that satisfies equation (13).

If the additional information is left out, the sentence has no comma:

Thus we found a number that satisfies equation (13).

(For 'that' as a relative pronoun, see also Rules 5.9 and 5.10 below.)

Rule 5.8 There is usually no comma before 'if'.

Example:

A graph is not 3-colorable if it contains a 4-clique.
Ein Graph ist nicht 3*-färbbar, wenn er eine* 4*-Clique enthält.*

However, if the 'if' clause comes first, it is usually separated from the main clause by a comma:

If a graph contains a 4-clique, it is not 3-colorable .
Wenn ein Graph eine 4*-Clique enthält, ist er nicht* 3*-färbbar.*

There are more exceptions to these rules than to Rule 5.7, which is why we are not discussing them here. Just keep in mind: don't put a comma before 'if' without good reason.

Rule 5.9 Non-defining relative clauses have commas.

Rule 5.10 Defining relative clauses have no commas.

In English, it is very important to distinguish between two types of relative clauses: defining and non-defining ones. This is a distinction you absolutely need to understand to write scientific texts, because mistakes in this area actually distort the meaning of your text!

It's probably easier to explain first what a *non-defining* relative clause is. A non-defining relative clauses simply gives additional information *that could also be left out* (or given in a separate sentence). For example, the sentence

> The WeirdSort algorithm, which was found by the famous mathematician John Doe, is theoretically best possible but difficult to implement in practice.

would be fully understandable if the relative clause were left out completely. It could also be rephrased as two separate sentences:

> The WeirdSort algorithm is theoretically best possible but difficult to implement in practice. [By the way,] WeirdSort was found by the famous mathematician John Doe.

This is what a non-defining relative clause is. *Non-defining relative clauses are always written with commas.* As a corollary we obtain that you cannot use 'that' in non-defining relative clauses (see Rule 5.7!). It would be wrong to write

> ~~The WeirdSort algorithm, that was found by the famous mathematician John Doe, is theoretically best possible but difficult to implement in practice.~~

A special case that warrants its own example is when 'which' is referring to the entire preceding sentence:

> Thus inequality (7) is true, which implies that the Riemann hypothesis holds.

As before, this is a non-defining relative sentence (it could be left out) and therefore needs a comma.

So let's discuss *defining* relative clauses next. A defining relative clause tells the reader *which specific item the main clause is talking about*. Leaving it out either changes the meaning of the sentence or renders it incomprehensible altogether. Consider the following example:

> The WeirdSort algorithm is difficult to implement in practice. In contrast, the algorithm that we suggest is very simple.

Here the relative clause 'that we suggest' cannot be left out – the remaining sentence would make no sense since the reader would not know which algorithm it is talking about. This is what a defining relative clause is. *Defining relative clauses are never written with commas.* Usually, you can use both 'that' and 'which' in defining relative clauses, although in many cases 'that' sounds better.

As a final example, consider the following sentence:

> For the elements in $\mathcal{B}$ which satisfy property (A), we know that

Table 5.1: Things you (usually) don't say

~~It holds (that) ...~~	We have ...	*Es gilt ...*
('Equation (5) holds.' is fine, though.)		
~~x fulfills property $\mathcal{P}$.~~	x satisfies property $\mathcal{P}$.	*x erfüllt Eigenschaft $\mathcal{P}$.*
~~in average~~	on average	*im Durchschnitt*
~~estimation~~	estimate	*Abschätzung*
~~composed number~~	composite number	*zusammengesetzte Zahl*
~~with the help of~~	using	*mit Hilfe von*
~~surely~~	clearly	*sicher, bestimmt*
~~monotonously increasing~~	monotonically incr.	*monoton steigend*
(Actually, in most cases 'increasing' is just fine.)		

equation (37) holds.

This sentence does not make a statement about all elements in $\mathcal{B}$, only about those satisfying property (A). The relative clause is *defining*. (Thus we could also use 'that' in place of 'which'.)

In contrast, if we add a comma the sentence reads

> For the elements in $\mathcal{B}$, which satisfy property (A), we know that equation (37) holds.

Now the relative clause is *non-defining* – it just mentions in passing that all elements in $\mathcal{B}$ satisfy property (A). The main clause states that equation (37) holds for *all* elements in $\mathcal{B}$. See the difference?

5.4 Things you (usually) don't say in English – and what to say instead

Table 5.1 lists some common mistakes and alternatives. The entries should not be taken as gospel – they don't necessarily mean that a given word or formulation is wrong under all circumstances (obviously, this depends a lot on the context). However, in nine out of ten instances the suggested alternative is the better word to use.

Chapter 6

Typography

6.1 Punctuation

Rule 6.1 Use opening (‘) and closing (’) quotation marks correctly.

In LaTeX, the closing quotation mark is typed like a normal apostrophe, while the opening quotation mark is typed using the French *accent grave* on your keyboard (the *accent grave* is the one going down, as in *frère*).

Note that any punctuation that *semantically* follows quoted speech goes inside the quotes in American English, but outside in Britain. Also, Americans use double quotes first. Oppose

> “Using ‘lasers,’ we punch a hole in … the Ozone Layer,” Dr. Evil said.

to

> ‘Using “lasers”, we punch a hole in … the Ozone Layer’, Dr. Evil said.

Rule 6.2 Use hyphens (-), en-dashes (–) and em-dashes (—) correctly.

A hyphen is only used in words like ‘well-known’, ‘3-colorable’ etc., or to separate words that continue in the next line (which is known as hyphenation). It is entered as a single ASCII hyphen character (-).

To denote ranges of numbers, chapters, etc., use an en-dash (entered as two ASCII hyphens --) with no spaces on either side. For example, using Equations (1)–(3), we see…

As the equivalent of the German *Gedankenstrich*, use an en-dash with spaces on both sides – in the title of Section 5.4, it would be wrong to use a hyphen instead of the dash. (Some English authors use the even longer emdash (—)

instead, which is typed as three subsequent hyphens in LaTeX. This emdash is used without spaces around it—like so.)

6.2 Spacing

Rule 6.3 Do not add spacing manually.

You should never use the commands \\ (except within tabulars and arrays), \␣ (except to prevent a sentence-ending space after Dr. and such), `\vspace`, `\hspace`, etc. The choices programmed into LaTeX and this style should cover almost all cases. Doing it manually quickly leads to inconsistent spacing, which looks terrible. Note that this list of commands is by no means conclusive.

Rule 6.4 Judiciously insert spacing in maths where it helps.

This directly contradicts Rule 6.3, but in some cases TeX fails to correctly decide how much spacing is required. For example, consider

$$f(a,b) = f(a+b,a-b).$$

In such cases, inserting a thin math space `\,` greatly increases readability:

$$f(a,b) = f(a+b,\, a-b).$$

Along similar lines, there are variations of some symbols with different spacing. For example, Lagrange's Theorem states that $|G| = [G : H]|H|$, but the proof uses a bijection $f\colon aH \to bH$. (Note how the first colon is symmetrically spaced, but the second is not.)

Rule 6.5 Learn when to use \␣ and `\@`.

Unless you use 'french spacing', the space at the end of a sentence is slightly larger than the normal interword space.

The rule used by TeX is that any space following a period, exclamation mark or question mark is sentence-ending, except for periods preceded by an upper-case letter. Inserting \ before a space turns it into an interword space, and inserting `\@` before a period makes it sentence-ending. This means you should write

```
Prof.\ Dr.\ A. Steger is a member of CADMO\@.
If you want to write a thesis with her, you
should use this template.
```

which turns into

Prof. Dr. A. Steger is a member of CADMO. If you want to write a thesis with her, you should use this template.

The effect becomes more dramatic in lines that are stretched slightly during justification:

Prof. Dr. A. Steger is a member of CADMO. If you

Rule 6.6 Place a non-breaking space (`~`) right before references.

This is actually a slight simplification of the real rule, which should invoke common sense. Place non-breaking spaces where a line break would look 'funny' because it occurs right in the middle of a construction, especially between a reference type (Chapter) and its number.

6.3 Choice of 'fonts'

Professional typography distinguishes many font attributes, such as family, size, shape, and weight. The choice for sectional divisions and layout elements has been made, but you will still occasionally want to switch to something else to get the reader's attention. The most important rule is very simple.

Rule 6.7 When emphasising a short bit of text, use `\emph`.

In particular, *never* use bold text (`\textbf`). Italics (or Roman type if used within italics) avoids distracting the eye with the huge blobs of ink in the middle of the text that bold text so quickly introduces.

Occasionally you will need more notation, for example, a consistent typeface used to identify algorithms.

Rule 6.8 Vary one attribute at a time.

For example, for WeirdSort we only changed the shape to small caps. Changing two attributes, say, to bold small caps would be excessive (LaTeX does not even have this particular variation). The same holds for mathematical notation: the reader can easily distinguish g_n, $G(x)$, $\mathcal{G}$ and G.

Rule 6.9 Never underline or uppercase.

No exceptions to this one, unless you are writing your thesis on a typewriter. Manually. Uphill both ways. In a blizzard.

6.4 Displayed equations

Rule 6.10 Insert paragraph breaks *after* displays only where they belong. Never insert paragraph breaks *before* displays.

LaTeX translates sequences of more than one linebreak (i.e., what looks like an empty line in the source code) into a paragraph break in almost all contexts. This also happens before and after displays, where extra spacing is inserted to give a visual indication of the structure. Adding a blank line in these places may look nice in the sources, but compare the resulting display

$$a = b$$

to the following:

$$a = b$$

The first display is surrounded by blank lines, but the second is not. It is bad style to start a paragraph with a display (you should always tell the reader what the display means first), so the rule follows.

Rule 6.11 Never use `eqnarray`.

It is at the root of most ill-spaced multiline displays. The *amsmath* package provides better alternatives, such as the `align` family

$$\begin{aligned} f(x) &= \sin x, \\ g(x) &= \cos x, \end{aligned}$$

and `multline` which copes with excessively long equations:

$$\begin{multline} \mathrm{P}\bigl[X_{t_0} \in (z_0, z_0 + dz_0], \ldots, X_{t_n} \in (z_n, z_n + dz_n]\bigr] \\ = \nu(dz_0) K_{t_1}(z_0, dz_1) K_{t_2 - t_1}(z_1, dz_2) \cdots K_{t_n - t_{n-1}}(z_{n-1}, dz_n). \end{multline}$$

6.5 Floats

By default this style provides floating environments for tables and figures. The general structure should be as follows:

```
\begin{figure}
  \centering
  % content goes here
  \caption{A short caption}
  \label{some-short-label}
\end{figure}
```

Note that the label must follow the caption, otherwise the label will refer to the surrounding section instead. Also note that figures should be captioned at the bottom, and tables at the top.

The whole point of floats is that they, well, *float* to a place where they fit without interrupting the text body. This is a frequent source of confusion and changes; please leave it as is.

Rule 6.12 Do not restrict float movement to only 'here' (`h`).

If you are still tempted, you should avoid the float altogether and just show the figure or table inline, similar to a displayed equation.

Chapter 7

Example Chapter

Dummy text.

7.1 Example Section

Dummy text.

7.1.1 Example Subsection

Dummy text.

Example Subsubsection

Dummy text.

Example Paragraph Dummy text.

Example Subparagraph Dummy text.

Appendix A

Dummy Appendix

You can defer lengthy calculations that would otherwise only interrupt the flow of your thesis to an appendix.

Bibliography

[1] Robert Bringhurst. *The Elements of Typographic Style*. Hartley & Marks, 1996.

[2] Tonsen Marc, Zhang Xucong, Sugano Yusuke, and Bulling Andreas. Labelled pupils in the wild (lpw): Pupil detection in unconstrained environments. https://www.mpi-inf.mpg.de/departments/computer-vision-and-machine-learning/research/gaze-based-human-computer-interaction/labelled-pupils-in-the-wild-lpw. Accessed: 28.04.2023.

[3] OpenCV. Contrast limited adaptive histogram equalization. https://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html. Accessed: 28.04.2023.

OST
Ostschweizer
Fachhochschule

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.