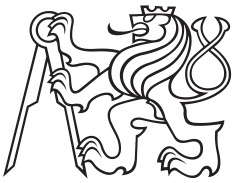
Bachelor's Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Circuit Theory

Technical support for fundamental psychovisual experiments

Technické zázemí pro základní psychovizuální experimenty

Petr Karlík

Supervisor: Ing. Karel Fliegel, Ph.D.
Field of study: Medical engineering and bio-informatics
May 2022

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Karlík Petr** Personal ID number: **491970**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Circuit Theory**

Study program: **Medical Electronics and Bioinformatics**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Technical support for fundamental psychovisual experiments

Bachelor’s thesis title in Czech:

Technické zázemí pro základní psychovizuální experimenty

Guidelines:

Provide a state-of-the-art review in the field of methods for performing fundamental psychovisual experiments. Focus primarily on the issue of experimental quantification of contrast sensitivity function (CSF). Using suitable programing environment and hardware, design tools for the implementation of selected psychovisual experiments. Verify the functionality of the system by experimenting with a group of observers.

Bibliography / sources:

[1] Pelli, D. G., Bex, P., Measuring contrast sensitivity, Vision Research, 2013.
[2] Lesmes, L. A., Lu, Z.-L., Baek, J., Albright, T. D., Bayesian adaptive estimation of the contrast sensitivity function: The quick CSF method, Journal of Vision, 2010.
[3] Watson, A. B., QUEST: A general multidimensional bayesian adaptive psychometric method, Journal of Vision, 2017.

Name and workplace of bachelor’s thesis supervisor:

Ing. Karel Fliegel, Ph.D. Department of Radioelectronics FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **26.01.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

Ing. Karel Fliegel, Ph.D.	doc. Ing. Radoslav Bortel, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

. _____ Date of assignment receipt

_____ Student’s signature

Acknowledgements

I would like to thank my supervisor Ing. Karel Fliegel, Ph.D. for guidance and valuable insight into the thesis format.

I would also like to thank FEE CTU in Prague, Department of Radioelectronics for providing all the equipment necessary to complete my thesis.

At last I would like to thank my family for patience and everlasting support.

Declaration

I hereby declare that this work is all my own work and I have cited all sources I have used in the bibliography.

In Prague, 15. May 2022

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 15. května 2022

Abstract

Characteristics of the human visual system play an important role in medicine, computer vision and image compression. Being able to measure these characteristic effectively is key for research development in these areas. In the theoretical part of this thesis we provide an overview of basic characteristics and introduce available methods for their measurement. After the overview the reader is then introduced with the problematic of contrast sensitivity function measurement. We discuss all the sources of error and later present a state-of-the-art algorithm for contrast sensitivity function estimation.

In the practical part we develop a device capable contrast sensitivity function measurement. We especially focus on its portability. To ensure results invariance with location change we provide a calibration tool. To prove devices portability we perform a series of test at two locations.

Keywords: Contrast sensitivity, Bayesian estimation, Psychovisual experiment, Psychometrics, humans visual system, Contrast, Contrast sensitivity function, Luminance

Supervisor: Ing. Karel Fliegel, Ph.D. Praha, Technická 1902/2, 166 27

Abstrakt

Základní charakteristiky lidského optického aparátu hrají důležitou roli zejména v medicíně, počítačovém vidění a kompresi obrazu. Schopnost efektivně měřit tyto charakteristiky je klíčová pro další výzkum ve zmíněných oblastech. V teoretické části práce poskytuji přehled základních charakteristik a dostupných metod pro jejich měření. Následně představuji čtenáři problematiku měření funkce kontrastní citlivosti. Diskutuji zdroje chyb měření a později představuji nejmodernější algoritmus pro odhad této funkce.

V praktické části vyvíjím zařízení schopné měření funkce kontrastní citlivosti. Zejména kladu důraz na přenosnost zařízení. Pro zajištění neměnnosti výsledků se změnou místa měření implementuji do zařízení kalibraci. Jako důkaz přenosnosti zařízení provádím sérii testů na dvou různých místech.

Klíčová slova: Kontrastní citlivost, Bayesovský odhad, Psychovisuálni experiment, Psychometrie, Lidský optický aparát, Kontrast, Funkce kontrastní citlivosti, Jas

Překlad názvu: Technické zázemí pro základní psychovizuální experimenty

Contents

Figures

Tables

Acronyms

- **CSF** - Contrast sensitivity function
- **SCSF** - Spatial contrast sensitivity function
- **TCSF** - Temporal contrast sensitivity function
- **CSS** - Contrast sensitivity surface
- **HVS** - Human visual system
- **CS** - Contrast sensitivity
- **API** - Application programming interface
- **AFC** - Alternate forced choice
- **cpd** - Cycles per degree
- **GUI** - General user interface
- **LED** - Light emitting diode
- **CRT** - Cathode ray tube
- **WLED** - White light emitting diode
- **CDF** - Cumulative distribution function
- **USB** - Universal serial bus
- **DDC** - Display data channel
- **LUT** - Look-up table
- **RAM** - Random access memory
- **OS** - Operating system
- **HDMI** - High-Definition Multimedia Interface
- **HD** - High definition
- **VGA** - Video graphics array
- **I2C** - Inter-integrated circuit
- **RMSE** - Root mean squared error
- **FEE** - Faculty of Electrical Engineering
- **CTU** - Czech Technical University
- **S*n*** - Subject n

Chapter 1

Introduction

In this thesis we first provide a comprehensive summary of basic psychovisual experiments in Chapter 2. We introduce key characteristics of the human visual system (HVS), provide overview of available measurement methods and discuss their use cases.

Chapter 3 introduce CSF measurement problematics. We choose CSF because compared to other characteristics there is not many user friendly devices available for CSF measurement. Also the research regarding CS in comparison to other HVS characteristics is rather sparse. To insentivise exploration of CSF's value we want to provide a portable and economically accessible device for its measurement. We try to optimize the trade off between devices accuracy and economical weight.

In the next sections we take a look at the evolution of available CSF measurement methods. We start with the simple ones like the Pelli-Robson chart[13] and work to the more mathematical methods like the Ψ-method [29]. Later we discuss the problems that occur while designing the device and experiment. At the end we provide a review of commercial devices for CS measurement.

We introduce the theoretical part of the chosen algorithm[2] in Chapter 4. We talk about its advantages and its limits. We also show a parametric form of CSF and later use this form to construct a statistical model of the observer which is used in the algorithm. Chapter 5 first presents the key parts of the software and hardware architecture. Here we argue the selection of both hardware and software tools needed to fulfill our goals 1.1. In the latter section we focus on the validation of our calibration probe. We use a reference device to asses the errors introduced by the low-cost device.

We perform multiple test at different locations to prove the portability and reliability of the device. Results of these test can be found in Chapter 6. We mainly focus on the comparison between measurements done in two distinct places on two very different monitors.

1.1 Thesis goals

The main goal of the thesis is to create a portable low cost device capable of measuring CSF. Especially we focus on the portability of the device we want to provide a fast calibration software which maximizes reliability of the results at different locations (conditions). Also we focus on the accessibility of all the parts need for the device both software and hardware. We use commercially easily available components and restrain from using any paid software. In summary we try maximize accuracy while constraining our economical resources.

We also want to give our work and educational dimension, we first focus on providing a comprehensive summary of HVS characteristics. Then we provide and in depth explanation of the used algorithm. We try to articulate its functionality better than the original papers can. At last we give the reader a possibility to verify the theory by performing a test himself. To simplify this procedure we provide manuals to help.

Part I

Theoretical part

Chapter 2

Basic psychovisual experiments

Psychovisual experiments aim to measure characteristics of the human visual system (HVS). These characteristics are important in many fields of research such as image compression [17], image enhancement [25][16], medicine [10] and many more.

Especially in medicine the relevance of these measurements is significant in diagnosis of many diseases. In this Chapter we will introduce few of these characteristics and some of their respective experiments.

2.1 Contrast sensitivity function (spatial CSF)

Since this function will be the main topic of the whole thesis I will introduce it briefly here and dive into measurement methods, history, avail in 3.1, 3.2. Contrast sensitivity function is in its base form a 2D function. It relates contrast sensitivity to spatial frequency. It can be expressed in many forms depending on the method used. Some of these forms are: 4-parameter truncated log-parabola, 3-parameter double exponential, 3-parameter log-parabola.

Spatial frequency

Spatial frequency in connection to CSF is mostly expressed in cycles per degree of visual angle. This is very handy as it directly relates the frequency of change to our visual system.

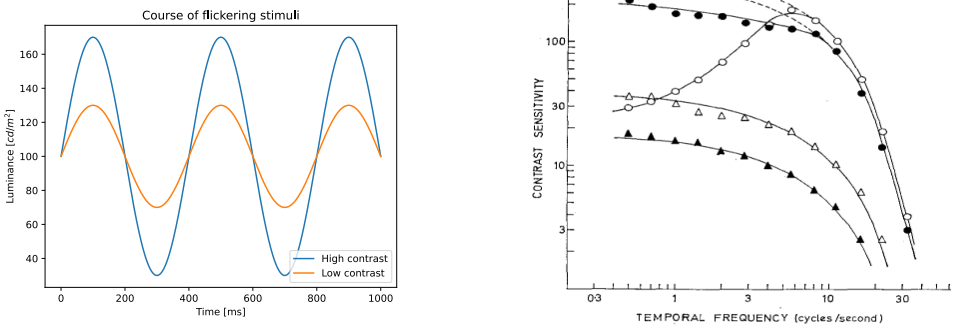
2.2 Temporal contrast sensitivity function

This function is derived from temporal contrast sensitivity which could be defined as a threshold exposure time needed for the stimuli to be detected reliably at a given contrast setting. So the TCSF relates contrast to temporal frequency.

Temporal frequency

This frequency is rather simpler than spatial frequency. It is a frequency of stimuli exposure to the tested subject. In simpler words this is the inverse of time we show the stimuli to the patient. We show the stimuli periodically.

The TCSFs form at different spatial frequencies is displayed on right part of 2.1, the left part shows standard stimuli flickering course.



(a) : Course of flickering stimuli **(b) :** TCSF

Figure 2.1: (a) Course of flickering stimuli and (b) Temporal contrast sensitivity functions at different spatial frequencies [28].

Measuring the TCSF

Authors of [32] used a rather classical approach. They first measured contrast by presenting a flickering stimuli to the subject whom then rotated a knob until they couldn't distinguish the flickering element from the background. In figure 2.2 we see a block diagram of their measurement. They tested nine different frequencies in random order. Adaptive Bayesian estimation approach could be used to estimate its shape if we reason some parametric form. I use this approach to estimate CSF.

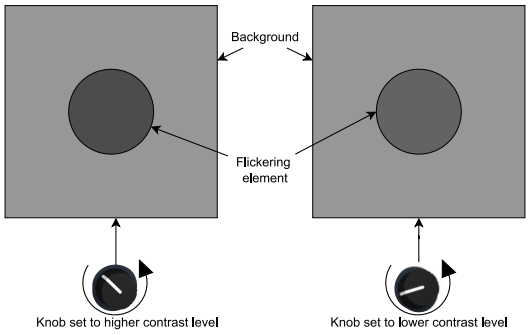
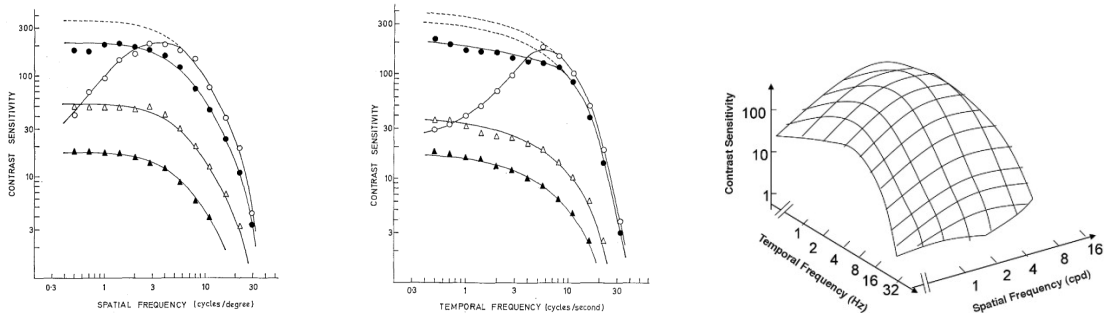


Figure 2.2: Block diagram of TCSF measurement.

2.3 Spatio-temporal contrast sensitivity function (surface) TSCSF

Since the temporal and spatial contrast sensitivity function are very connected it naturally makes sense to combine them into one function and define it as a function of both spatial and temporal frequencies. Being able to measure this function accurately and also in acceptable time could bring more data for further analysis and ultimately lead to deeper understanding of human visual system. Combining both of these functions into one test could bring substantial increase in speed of gathering the necessary data needed for correct estimation of TSCSF than measuring (S)CSF and TCSF separately. Robson [28] was probably the first to come with the mixed measurement. He measured separately spatial CSF on different temporal frequencies and temporal CSF on different spatial frequencies. [7] then measured the surface. All their results summarized in figure 2.3



(a) : SCSF **(b) :** TCSF **(c) :** CSS

Figure 2.3: Spatial (a) and temporal (b) contrast sensitivity functions at different levels of spatial and temporal stimuli [28] and (c) Contrast sensitivity surface [7].

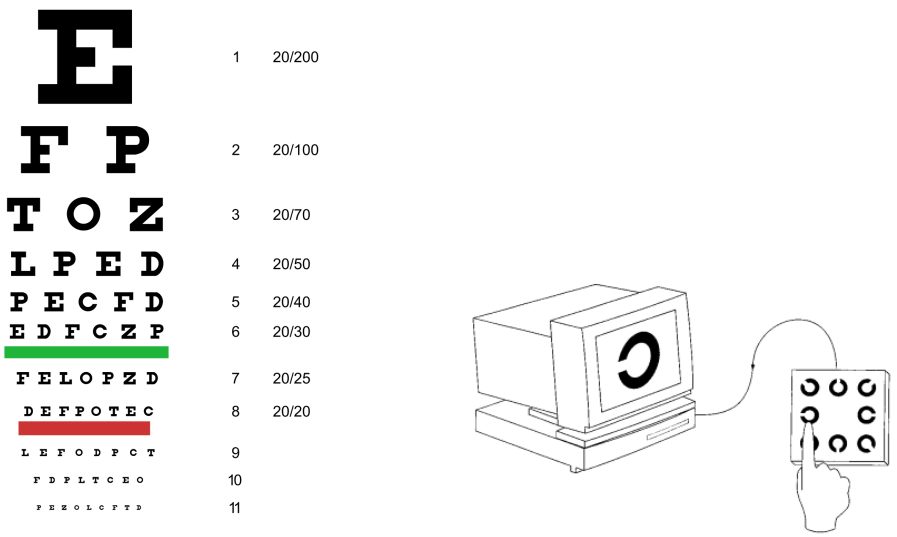
2.4 Visual acuity

Visual acuity is described as the ability to differentiate between two objects in space with high contrast relative to each other. Mostly we define it as the smallest object we are able to perceive from a given distance. It is the basic visual characteristic doctors measure.

Measurement methods

The basic one is the Snellen chart. It shows incrementally smaller numbers or letters. The subjects task is to recognize the smallest letter or number he can. The fractions on the side are used as a measure of acuity. Let us break them down. The numerator stands for the distance from the chart. Standard value is 6m or 20 feet in our case. The denominator is a distance from which a standardized observer would be able to read the line. So 20/20 is a line that a normal person should be able to read. The letter with 20/200 would be read by a normal observer from a distance of 200 feet. Also if you are unable to read this line from 20 feet, legally you are classified as blind. Left side of 2.4 shows a Snellen chart.[5]

Another very popular method is the Freiburg visual acuity test. This test utilizes circle with a gap somewhere around it.The acuity is given by the size of the gap. Subject then determines the gaps direction as shown on the right side of 2.4.[22]



(a) : Snellen chart **(b) :** Freiburg test course

Figure 2.4: (a) Snellen chart [1] and (b) Freiburg visual test demonstration [22].

Chapter 3

Measuring the CSF

In this Chapter we provide a comprehensive summary of all the possible problems related to CSF measurement. We also argue its importance and provide an overview of available measurement methods. Later we introduce the errors associated with hardware, software and human intervention which plays a big role in all the of psychometrics.

3.1 Why measure the CSF

CSF plays very important role in diagnosis of different illnesses such as Parkinson's disease [4], Alzheimer's disease [9], [21] concluded that it could be a relevant marker of incoming severe retinal disruption before pathological problems occur, [27] established that CSF is an important marker for evaluating glaucoma patients ability to undertake daily life tasks. Many other studies also proved CSFs importance in diagnosis of many defects both visual and other.

3.2 Methods for measuring the CSF

In this section we will discuss different methods for measuring the contrast sensitivity functions. Starting from the oldest methods like Pelli-Robson chart[13] or Arden gratings[14] to the newest ones based on adaptive parameter estimation using statistics/machine learning methods[2][31]. The first two methods I will present are quite simple but very powerful due to that. I introduce

these methods because they show two of the most common stimuli used when evaluating the CSF.

3.2.1 Pelli-Robson chart

The Pelli-Robson chart[13] is a test developed to measure contrast sensitivity (CS). Thus it is incapable of estimating the CSF. But due to its speed it managed to stay relevant for fast evaluation of patients general CS. Its usability proved crucial when reviewing the CS of infants as they are incapable of sophisticated feedback.

How it works?

Pelli-Robson chart is basically a Snellen chart but instead of alternating the size of the letter we vary the contrast of the presented letters while keeping their size constant. In figure 3.1 a typical Pelli-Robson chart can be seen. The experiment is subjective as the patient can guess the answer. This phenomenon occurs in every method we present. Later we will see some more sophisticated method trying to deal with this fact at least at some level.

Figure 3.1: Pelli-Robson chart[8].

3.2.2 Arden gratings test

Arden[14] proposed a different approach to stimuli generation. He used sine wave grated stimuli. He changed the frequency of the sine wave which generated the grated stimuli, we can observe this in Figure 3.2. Doing this he was able to correlate the frequency to the current patient anamnesis. He found that indeed the CS plays a role in diagnosis of independent ailments.

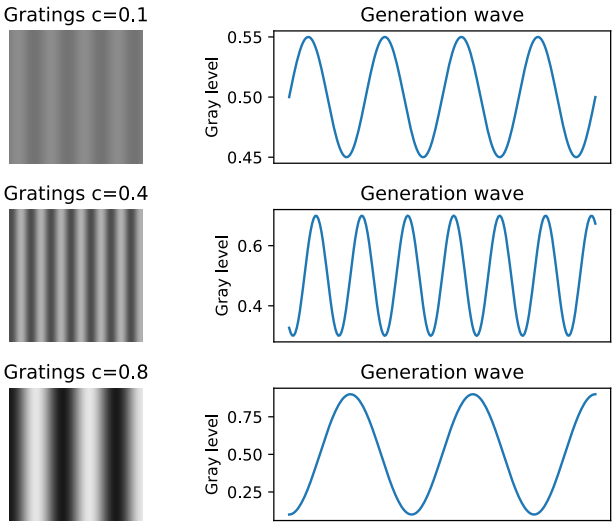


Figure 3.2: Sine gratings generation.

3.2.3 Ψ method

Leonid L. Kontsevich and Christopher W. Tyler came up with this method in 1999 [29]. They weren't the first ones to try an adaptive method to evaluate such psychometric function but we could say they were the most successful, since most newer methods use it as a reference/comparison [2], [15].

The form of the CSF in Ψ method

Authors estimated two parameters of the CSF. These two parameters define the psychometric function and the CSF. Indeed there is a difference between the two. The CSF is the actual curve we connect to the tested subject and the psychometric function is a probability model used mainly when estimating the function. This difference will be clear when we introduce the method used in my device.

Bayesian adaptive estimation

This approach uses the bayes rule to update the probability that the current estimation is the correct one. We update this probability after every response given by the tested subject. We use entropy as a cost function to determine which presented stimuli will allow us to gather the most information in the next cycle, thus calling it adaptive.

Originally [30] came up with this approach for estimating single parameter psychometric function. Later Andrew B. Watson [31] generalized this approach for experiments which require:

- More than one psychometric function parameter
- More than one stimulus parameter
- More trial outcomes (patient responses)

Stimuli

When testing their method the authors opted for the Gabor stimuli in a 2AFC task. Gabor stimuli is basically an Arden plating but it is filtered by a Gaussian. This softens the edges and makes the whole stimuli round.

The stimuli had a spatial frequency of 2cpd. Here we run into the main limit of the Ψ method. The whole test is conducted on the same spatial frequency only the contrast is changing. This is also the reason why we define the intensity of the stimulus x as a scalar not a vector. Later we will explore methods which stimuli vary in both intensity and spatial frequency.

Note that because this test only predicts the threshold and slope. Using sole spatial frequency is reasonable as the authors only want to predict these two parameters of the function not the CSF itself.

Ψ method - conclusion

The Ψ method definitely brought a new approach for measuring the CSF parameters. It came with the idea to perform a one-step ahead search using the entropy as a cost-function. Further methods build on this idea. Its biggest drawback is the fixed spatial frequency of the stimuli which makes it impossible for this method to successfully estimate the whole CSF not just two of its parameters.

Its stability could also be further improved by using an AFC test with more choices, although the authors prove it's reasonably stable even with smaller amount of trials. This method opened the gates to parametric estimation of the CSF.

3.2.4 qCSF

The qCSF method is again a method which uses the Bayesian adaptive framework. It builds at the foundation laid by the Ψ method. As this will be the method we implement on our device we will only mention it here and explain it in Chapter 4. In figure 3.3 we see timeline of the methods development.

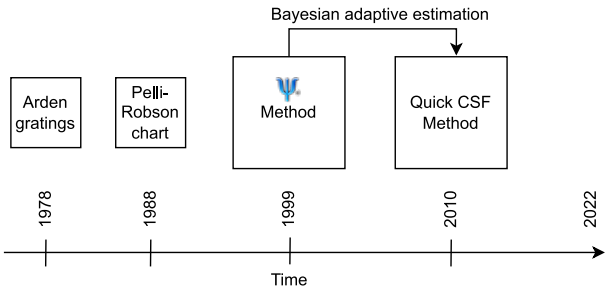


Figure 3.3: Methods evolution timeline.

3.3 Statistical and technological influences on CSF testing

When measuring virtually anything it is important to be aware of the test limits and what could influence the results of our measurement. I will discuss the most significant sources of error and how to subordinate them. The sources can be roughly categorized into technological, meaning that their origin is inside the device which governs the test, such as display, processors and more, and statistical meaning their primary source is sometimes not known but can be characterized by some probability.

3.3.1 Influences originated in technology

The technological influences are worth mentioning only when considering tests that are presented on some digital device since the paper tests mostly don't suffer from these effects or they are not accurate enough to be worth mentioning them.

Display luminance

Displays calibration plays a huge part in the test reliability. In order for the test to be reliable it has to yield similar results for the same tested subject when independent measurements are made. This is especially important in psychometrics as the test is often divided into sessions. So it is very important to provide the same testing environment for the tested subject in order not to skew the results of the test.

The main concern is the displays luminance range. If we want the test to be reliable we need to ensure luminance of the presented stimuli is in fact the one we need. This raises a big need to calibrate the used display accordingly. Later we will solve this problem by estimation the transfer function of the display and calibrating its white point to luminance we need. This luminance should be the same for every used display otherwise the measurement will be biased. We will follow [2] and set this luminance to $120cd/m^2$.

Displays refresh rate

This parameter is important when testing TCSF. First we should measure the real refresh rate of the monitor to know the exact frequency and calibrate our test accordingly. It is very important to not break the Nyquist theorem as this would also tamper with the results.

Influence of display parameters on stimuli

Since we measure spatial frequency in cycles per degree of visual angle the perceived frequency changes with the subjects distance from the screen. Also we have to take into account that the display is basically a discrete grid. First let us derive how many pixels are 1 degree of visual angle. From this number we can easily determine the limits of the test.

First let us derive the size of pixel in cm.

$$c = \frac{W}{r} \tag{3.1}$$

Where:
W - width of the screen [cm]
r - resolution of the screen [px]
Now size in cm of visual angle in degrees on the screen is:

$$b = \tan(\varphi) \cdot d \tag{3.2}$$

Where:
φ - visual angle in degrees
d - subject distance from the display [cm]
Now easily the size of visual angle in pixels is its size in cm divided by the size of one pixel in cm.

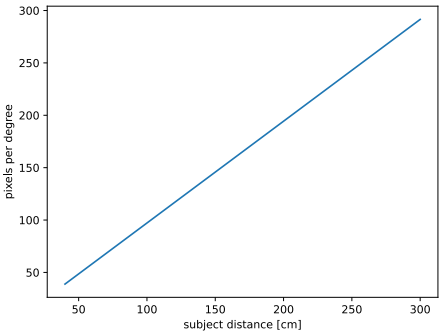
$$p = \frac{b}{c} \tag{3.3}$$

For example my screen has $W = 34.5cm$, $r = 1920$ and lets take $d = 0.5m$ then:

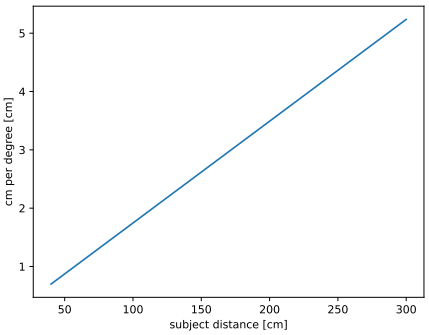
$$p = \frac{\tan(1)0.5}{\frac{0.345}{1920}} \approx 49pix \tag{3.4}$$

Sizes both in pixels and in centimeters are in 3.4.

For distance of 0.5m 1 degree of visual angle is 49 pixels on my screen. This means if we have a 400×400 stimuli we will fit only 3 periods of 0.5cpd stimuli. This calculation then easily infers the conversion to cycles per pixel. For 1cpd it would be 1/49 cycles per pixel as full cycle is 49 pixel then one pixel makes 1/49 of the whole.



(a) : Number of pixels per 1 degree to distance.



(b) : Number of centimeters per 1 degree to distance.

Figure 3.4: Sizes of 1 degree of visual angle at different viewing distances in (a) pixels (b) centimeters.

If we want to measure up to 30cpd we will encounter a problem as one period is assigned a little more than one pixel. There are more solutions to this. The obvious one would be to increase the distance and that is how I am going to solve this problem later. Another possibility would be to change the display.

Inconsistent subject distance error is caused by the edges of the stimuli actually being further away from the subject. Since the units of cycle per degree are dependent on the subject's distance thus the spatial frequency further away from the center is burdened by a systematic error. The error is highest furthest away from the center of the presented stimuli. We will use square stimuli the max distance will be half the diagonal of the square. Also note since the used stimuli is filtered by a Gaussian window the effects of this error are minimal since the center of focus for the tested subject is the center. In 3.5 we can see how the error is related to the subject distance.

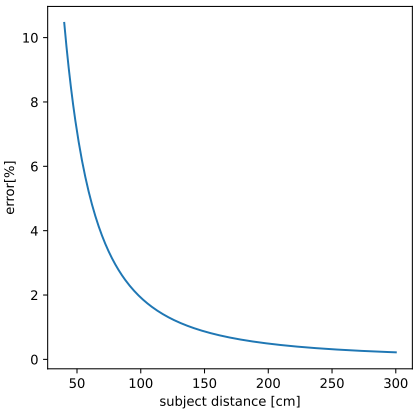


Figure 3.5: Subject distance error 13,7cm away from the center.

Figure 3.5 is display dependent so for bigger screens this error will be higher. Since errors nature is purely systematic it can be compensated. However we will not be implementing the compensation as the main focus is on the center of the display where this error doesn't occur.

3.3.2 Influences originated in statistics

There are two major statistical influences. First one is the miss rate of the patient, second one is the correct guess probability.

Miss rate

The miss rate is very straightforward. It can be defined as the probability that the tested subject won't detect a stimuli which according to his real CSF or another tested function, he should. Normally this probability is very low but in a badly designed test it could get higher. It is fair to note that the ψ and the qCSF method are not very sensitive to this. These methods will suppress the error quite quickly as tested in their respective papers.

Correct guess probability

We already mentioned this when describing different methods test parameters. As in most psychometric tests we use forced choice. This leads to errors generated by the patient being forced to take a guess and guessing correctly. This risk can be minimized by using tests with more options. More formally using higher n in n-AFC tests. The probability of answering correctly is $p = \frac{1}{n}$. This only works if all the choices available have the same probability to be selected. For example if we were to use number 0-9 in 10×10 pixel grid some numbers will occupy less pixels than other numbers. This will lead to certain numbers being easier to spot than other thus biasing the choice. We need to be aware of this when designing our stimuli set.

3.4 Currently available testing devices

There are only a few devices available. All of them are standalone devices designed to be used in a doctors office/clinic. None of the manufacturers provides any prices probably because these devices are sold individually to ophthalmology clinics and are expensive. There also exist many printed versions of either Pelli-Robson chart of some modification of them. Now we will introduce and compare the three most relevant devices available.

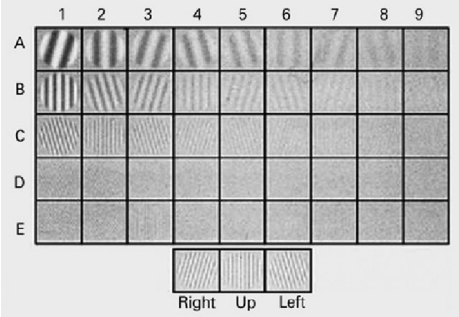


Figure 3.6: Similar grating test to VectorVision [19]

3.4.1 CSV1000 & CSV2000

These two devices are manufactured by the same company, VectorVision[1]. They both offer a measurement only on given contrasts (8 standardized levels) and four spatial frequencies (3,6,12,18 cpd). These devices provide many other visual test and also allow the operator to combine them. For example it is possible to measure visual acuity on different contrast levels of the presented letters. For a doctor this could be very informative, but all these tests lack statistically signifanct amount of trial and most importantly they don't perform any statistical estimation of the CSF. The devices are only suitable for a rough evaluation of subjects vision, not a comprehensive inference about their transfer function such as CSF.

Calibration

Both the devices continuously measure the monitors brightness and adjust it to $85cd/m^2$. This ensures great repeatability of all the offered tests. Since the monitor is included with the device the calibration can be properly adjusted to the exact display. This allows them to achieve very stable results. As far as the subjects distance from the apparatus, manufacturer doesn't provide any information but based on the quality of the apparatus it is safe to assume this is counted upon.

How they measure contrast sensitivity?

VectorVision provides multiple CS tests most of them are yes-no based. Mostly in psychometric function estimation the yes-no tests are rejected as insufficiently controlled. Worth mentioning is the letter contrast test which is basically a digitalized Pelli-Robson chart and the gratings test where the subject is presented with multiple gratings of the same spatial frequency but gradually decreasing contrast as in 3.6. Subject is then asked to identify which contrast level they can still see.

[1] https://vectorvision.com/

3.4.2 OCULUS Binoptometer 4P

This manufacturer[2] presents a different approach they provide the standard microscope like binocular device. The subject looks into the device and is presented with a task. Again they provide a wide variety of visual tests ranging from contrast, acuity, color etc. For contrast they use the same stimuli Freiburg [22] uses in his acuity test, only the circles gap doesn't change only the contrast of the whole circle changes. The device is controlled by a separate tablet on which the doctor manages the whole test. As far as the results go the producer doesn't provide much info on how the results are presented but given the nature of the tests it is safe to assume no inference is being applied to the results of the tests and the operator is only presented with the raw data.

Technical parameters

OCULUS Binoptometer uses a small color display which is 800×600 pixels. The brightness of the visual field presented is quite high, $300cd/m^2$. The manufacturer doesn't specify anything about a display calibration.

We will compare the presented devices plus two other devices in the next table. The additional devices manufacturers don't provide much info, thus the reason we only mention them here. We put "∘" into to cell if the device has the feature. We put "×" if the feature is unavailable. We leave the cell empty if the information is unreachable. Now we will explain the features.

1. Threshold - If it provides the threshold CS.
2. Graph - Graphs the test results.
3. Length of the test.
4. Digital or printed.
5. Calibration - If the device self calibrates luminance.
6. Letters - Are letter stimuli available.
7. Comparison with normalized values.

[2] https://www.oculus.de/us/products/visual-test-equipment/binoptometer-4p/highlights/#produkte_navi

	Threshold	Graph	Length	Digital	Calibration	Letters	Comparison
CSV1000	○	○		○	○	○	○
CSV2000	○	○		○	○	○	○
OCULUS Binoptometer 4P	○	×	fast	○		×	
M&S-tech HACSS	○	○	10min	○	○	×	○
M&S-tech Sine gratings	○	○	×	○		×	○

Table 3.1: Table of devices and its features
(○ - feature present, × - feature unavailable, " " - information unreachable)

3.4.3 Available implementations

There exist many toolboxes for psychometrics (*psychtoolbox*- MATLAB, python). Most of these toolboxes provide good background for simple testing but do not provide the methods we need. We mention two implementations here. Both of them are open source thus freely accessible. Note there should also exist MATLAB implementation of qCSF method by the authors[2]. Unfortunately the code was unreachable.

1. qCSF - [6]
2. bayesian adaptive estimation - general implementation in Mathematica available in [31]

Implementation 1 provides source for qCSF method in python. Apart from the code for the estimation algorithm it provides a GUI and a simple test. However the test has a few flaws, the size of the stimuli they use is very small thus causing significant error for standard displays. Also the generation of the stimuli was insufficiently controlled in relation to display calibration. So we decided to implement calibration, stimuli generation, GUI and adapt it to their implementation of the estimation method.

Implementation 2 is a general QUEST+. QUEST+ is basically a general form of the qCSF algorithm. The API they provide is very nice but Mathematica is a platform which would require to be used either on a desktop computer or the whole thing be re-implemented in another language compatible with the Raspberry Pi we plan to use. So we decide to not use this implementation at all.

3.4.4 Displays in computer based tests

I will present a table of display specifications used in two different papers both using the qCSF algorithm. One is the authors paper [2] second one is a qCSF done with 10-AFC task

[15].

ref.	Type	Refresh rate [Hz]	Diagonal
[2]	CRT	85	17 inch
[15]	LED	60	32 inch

Table 3.2: Table of used displays

None of these displays are standard. The LED display is a professional monitor designed for controlled experiments it uses WLED technology. CRT displays are obsolete now they can provide good monochromatic bit depth which is handy for our measurement. We will not be using any display like these two as one of the main goals is to provide a device which can be plugged to common monitors.

Chapter 4

The implemented method and the course of experiment

In this Chapter we will introduce the implemented qCSF [2] inference method and explain how the whole experiment works. We will omit the hardware setup of the experiment and just focus on the theory here. We will talk about the concrete implementation and hardware setup in the next Chapter.

4.1 The representation of CSF

In this section we will present both the parametric form of the function we are trying to estimate and also the statistical model of the psychometric function which will further be used to model the responses and perform the inference.

4.1.1 Parametric form of CSF

We characterize the CSF using 4 parameters. The peak sensitivity γ_{max}, the peak spatial frequency f_{max}, bandwidth at half maximum β and the truncation δ which is necessary due to the low pass behaviour of the CSF at lower spatial frequencies. We define the untruncated form of the function in log units as follows:

$$CSF(f) = \log_{10}(\gamma_{max}) - \log_{10}(2)\left(\frac{\log_{10}(f) - \log_{10}(f_{max})}{\log_{10}(2\beta)/2}\right)^2 \tag{4.1}$$

The reasoning behind this concrete parametrization is mainly the interpretability of the parameters. For example [23] also evaluates the difference of Gaussians which exhibits very similar fit but its parameters are not as interpretable as the ones of untruncated log-parabola which the qCSF algorithm implemented here uses. The next figure shows the function form and the parameter influence on it.

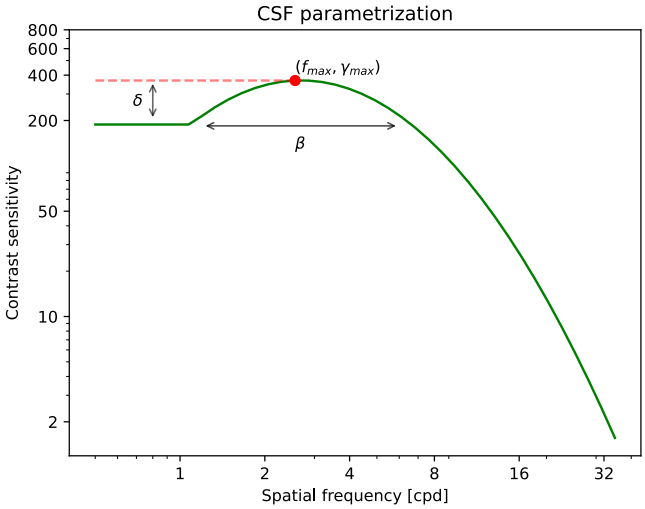


Figure 4.1: The parametrization of CSF in log scale

4.1.2 The psychometric model used for estimation

The probabilistic model is needed for the inference algorithm. In most psychometric tests we try to relate only one variable for example only contrast. Thus we can use expand the one dimensional value we are trying to estimate by a probability axis. Note that this is natural as when we estimate a threshold we expect to be more uncertain the closer we are to the actual value. In our case we are estimating a relation between two variables thus our probabilistic model will be a function of these two variables. Our variables or more accurately stimulus parameters are contrast sensitivity and spatial frequency. Since we will later discretize the space of all possible stimuli (combinations of these two parameters) we will need to have a way to assign a probability to each possible stimuli. Note that the model will inherently be bound to some actual CSF defined by the parameters mentioned above. Further it will be convenient to put these parameters in a vector we will call this vector $\boldsymbol{\theta}$.

Reasoning behind choosing the model

Let us again use the threshold estimation example. As mentioned above it makes sense for the model to be uncertain near the threshold and very certain further. Functions which exhibit similar behaviour as a logistic sigmoid functions are the perfect candidates in this case. In psychometric we mostly use what is called a weibull distribution. In statistic this function is the CDF of weibull distribution. Here is the logistic sigmoid and weibull function in one plot.[12]

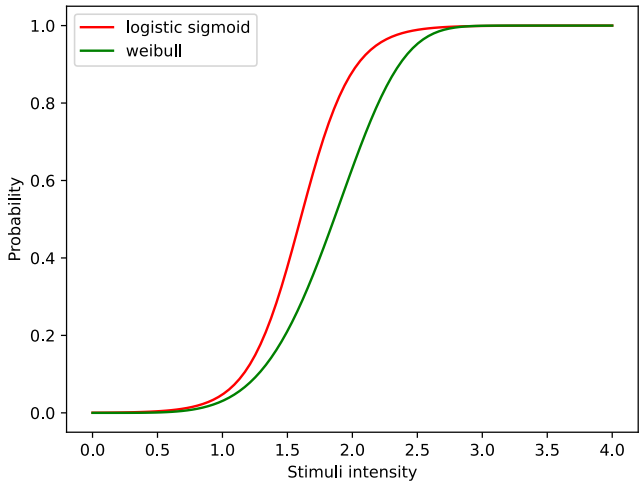


Figure 4.2: Logistic sigmoid and Weibull function

We see that both of these function exhibit exactly the behaviour we need. So now to the CSF model. We would like to use something similar but we need this behaviour around every point of the CSF. We can imagine that every point on the CSF is a threshold. Also we can't go very steep around the CSF as this will mess up the algorithm we will talk about this later. We define the psychometric function as [6]:

$$\Psi_{\boldsymbol{\theta}}(f,s) = 1 - \frac{1-\frac{1}{m}}{1+e^{(CSF_{\boldsymbol{\theta}}(f)-\log(c))/\sigma}} \tag{4.2}$$

Where:
c is sensitivity (inverse of contrast threshold)
f is spatial frequency
m defines the alternate forced choice m parameter, basically it is the probability of a correct guess
σ is the deviation or slope which defines how fast the probability falls off around the contemporaneous CSF, in our case $\sigma = 0.25$

Note that the subscript $\boldsymbol{\theta}$ symbols attachment to a concrete parameter vector $\boldsymbol{\theta}$.

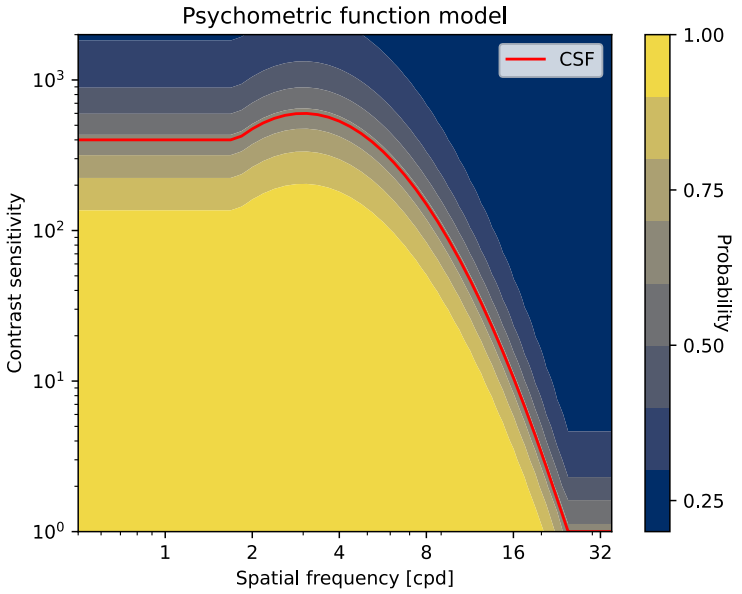


Figure 4.3: The psychometric function defined by 4.2 for $m = 4$

4.2 The estimation algorithm

The algorithm combines Bayes statistics with iterative approaches most commonly found in machine learning algorithms such as AdaBoost or Neural nets. It is based on the assumption that the psychometric model used sufficiently describes the probabilities of user response. Fundamentally it can be divided into two parts:

1. The inference part
2. Adaptive selection of next stimulus values

The inference part utilizes Bayes rule to update posterior knowledge about the parameter space. By shaping the probabilities defined over the whole parameter space we harden our certainty about the actual correct unknown parameters.[2]

The adaptive part tries to further accelerate the whole experiment by maximizing information gain, this will be explained further. First let us define the probability of getting a response r_t where t is either *correct* or *incorrect* given a vector of parameters $\boldsymbol{\theta}$ and a stimulus s.
The probability of correct response is defined as:

$$p(r_{correct}|\boldsymbol{\theta}, s) = \Psi_{\boldsymbol{\theta}}(s) \tag{4.3}$$

And incorrect:

$$p(r_{incorrect}|\boldsymbol{\theta}, s) = 1 - \Psi_{\boldsymbol{\theta}}(s) \tag{4.4}$$

Note that following the Ψ method, since our parameter and stimulus space is discrete, all these values could and should be computed before the experiment and used as a LUT.[29]

Let i denote the i-th trial. We then calculate the probability of getting a response r_t at trial i as:

$$p_i(r_t|s) = \int_{\boldsymbol{\theta}} p(r_t|s, \boldsymbol{\theta}) p_i(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{4.5}$$

Since we work in a discrete space we can further simplify the integral to a sum.

$$p_i(r_t|s) = \sum_{\boldsymbol{\theta}} p(r_t|s, \boldsymbol{\theta}) p_i(\boldsymbol{\theta}) \tag{4.6}$$

This probability tells us how likely a response r_t is given a stimulus s. It plays a significant role in Ψ-method[29] we will just use to normalize 4.7.
Now let us use the Bayes rule to define the posterior given a stimulus s.

$$p_i(\boldsymbol{\theta}|r_t, s) = \frac{p_i(\boldsymbol{\theta}) p(r_t|s, \boldsymbol{\theta})}{p_i(r_t|s)} \tag{4.7}$$

This posterior tells us how likely a vector $\boldsymbol{\theta}$ is for the current stimulus s. In the next step we will try to find the stimulus s which will introduce the most information to the calculated posterior. Also note that the denominator is only the sum of the numerator. Sometimes this is called the probability of the data. In the equation it has no direct meaning it only ensures the numerator sums to 1. In fact formally it shouldn't be the sum of the numerator but an integral like in 4.5. We obviously simplify to discrete case thus changing the integral to sum. Now let us evaluate the predicted posterior probability space using entropy.

Reasoning behind using entropy is quite simple. Our goal is to iteratively update our knowledge about $p_i(\boldsymbol{\theta})$. So we need a way to evaluate how good a certain posterior is. Entropy is widely used in statistics and machine learning. It finds its usage when as a cost function when constructing decision trees[26], or in neural nets[18] and many other areas.

Entropy is a great way to evaluate a probability space as its values are lower as more the space is condensed, meaning it favours certain values, the lower the entropy is. So now we have a tool to measure "condensness" or spread of probability space. But this is exactly what we need since our goal is precisely to choose the posterior which is the most condensed. This posterior is the one where we are most certain about some parameters. We can find this posterior by minimizing the entropy as a function of stimulus s. But this is rather slow as [20] suggests so we will use a speed up, we explain further.
Here would be a good time to say we don't have any correlation between the parameters we

are converging to be sure about and the actual correct ones. But we assume the model to be explicit enough to make this link.
We will need entropy for a binary distribution which is defined as follows

$$h(x) = -p\log(p) - (1-p)\log(1-p) \tag{4.8}$$

Entropy is very intuitive so it should be clear that instead of minimizing the expected entropy we can reformulate this problem as maximizing the difference between the entropies in the current and next trial. This measure is known as information gain. Authors of [20] exploited few fundamental properties of information gain such as symmetry and parametrization invariance to derive a formula for information gain. I will not derive the whole thing here but just introduce the final steps.

$$I_t(\boldsymbol{\theta}, s) = h\left(\int p_t(\boldsymbol{\theta})\Psi_{\boldsymbol{\theta}}(s)d\boldsymbol{\theta}\right) - \int p_t(\boldsymbol{\theta})h(\Psi_{\boldsymbol{\theta}}(s))d\boldsymbol{\theta} \tag{4.9}$$

Computing these integrals is very heavy(same as computing all the entropies and then finding their minimum as mentioned above) so we turn to a Monte-Carlo approximation. This is the main difference between this method and the Ψ-method[29]. We draw N random i.i.d samples from the prior distribution $p_i(\boldsymbol{\theta})$. We can then estimate the information gain as:

$$I_t(\boldsymbol{\theta}, s) \approx h\left(\frac{1}{N}\sum_j \Psi_{\boldsymbol{\theta}_j^i}(s)\right) - \frac{1}{N}\sum_j h(\Psi_{\boldsymbol{\theta}_j^i}(s)) \tag{4.10}$$

Obviously we omit the prior which occurs in the integral form as we sample from its respective distribution. We can now find the global maxima of this function to obtain the stimulus s which maximizes information gain. Compared to the Ψ-method[29] this step has immense advantage as it allows us to only compute 4.7 for the best stimulus s which we already know. Now we can run the trial to gather a response r_i and then we use 4.7 to obtain the posterior.

Extracting the parameters from the probability distribution

Now it should be obvious that a each posterior somehow defines a concrete CSF because we need this parameter estimate to perform the next trial namely to compute 4.10. There are many ways to extract the CSF parameters from the posterior (e.g prior) probability, for example we could use the maximaly probable values. [2] opted for marginal means so that's what we are going to use. Once we obtained the parameter estimates we can return to calculating 4.6 and to perform the next trial.

Termination condition

There are two options how we could terminate the experiment:

1. Define some heuristic rule
2. Run a defined number of cycles

In psychometrics it is common to go the easy way and just set a fixed number of cycles. Heuristic methods tend to use some threshold error as a termination condition this is error could be hard to compute in psychometrics.

Also in psychometrics the length of the test is very important as the patients tend to loose focus after some time of testing. Thus their responses become biased. With heuristic methods we can only speculate about the time it is going to take. Of course we could combined both of these approaches and limit the number of cycles again.

There are studies which suggest better performance for heuristic methods such as:
[3] they reported better performance in controlled termination condition for 2-AFC tasks, but the benefit was too small and mostly occured with low number of iterations where the estimation wasn't very accurate anyway. So fixed number of iterations is well justified.

4.3 The stimuli

As [2] we used a Gabor stimuli which is a sine grating smoothed by Gaussian window. As tackled previously to avoid problems associated with high spatial frequencies which can result in only a few pixels per period or low frequency stimuli not fitting enough periods into the window, we opted for larger stimuli 1000×1000 pixels. The luminance profile of the stimuli was defined as follows

$$s(x,y) = L_0 \left(1 + c \cdot \sin\left(2\pi f(x\cos\theta + y\sin\theta)\right) \cdot e^{\frac{-(x^2+y^2)}{2\sigma^2}}\right) \tag{4.11}$$

Where:
L_0 is the mean luminance, should be around $59 - 60 cd/m^2$. Since we calibrate the display white to $120 cd/m^2$ and leave the black to the minimum display can provide. L_0 depends on both of these values as it is the mean of them.
c is the contrast
f is the frequency, generally converted to pixels as f_{cpd}/g where g is the size of one degree of visual angle in pixels
θ is the angle by which the stimuli is rotated
σ is the standard deviation fixed on 1.8 degrees of visual angle

We use the Gaussian window to focus the subjects attention to the center of the screen.

Part II

Practical part

Chapter 5

Software and hardware realization

In this Chapter we will describe the architecture of the software and hardware. We reason these choices to accommodate our main goal of creating a portable device capable of creating reliable conditions for CSF measurement. We will also discuss the hardware limits and especially the calibration which the whole device requires. We also present a measurement performed with two independent calibration devices. We use a very precise measurement device to review the performance of a commonly accessible calibration probe and to asses reliability of the implemented calibration procedure.

5.1 Software architecture

In this section we will talk about the key software tools used for development. We will argue their choice and discuss their advantages. Next we describe the programs architecture 5.1.2 and introduce the main parts of the source code.

5.1.1 The development tools used

The whole program is written in Python. We chose Python mainly because Raspberry Pi's 5.2 support for running Python programs is very good. Also Python enables us to write, debug, run the code on PC and transfer it to Raspberry Pi without much modification. Although some problems occurred especially when setting up packages on Raspberry Pi. Most of

these problems were solved by installing these packages directly not through some package management system like *pip* or*anaconda*.

5.1.2 The program architecture

We can divide the program into three main blocks.

1. Initialization life-cycle
2. Calibration life-cycle
3. Experiment life-cycle

The initialization life-cycle consists of a simple GUI which is used to gather data about the experiment to be conducted. The data consists of:

1. Experiment name - will be used as the file name if you wish to save the experiment (default: 'results_log')
2. Number of trials
3. Screen diagonal in inches - doesn't support displays with different aspect ratio than 16:9 (the most common)
4. Display resolutions (width, height in pixels). Should already be filled as we can fetch the info from the computer so it only needs to be checked
5. Subject distance - distance of the tested person from the display

Calibration life-cycle

A simplified diagram of the calibration life-cycle is in figure 5.1

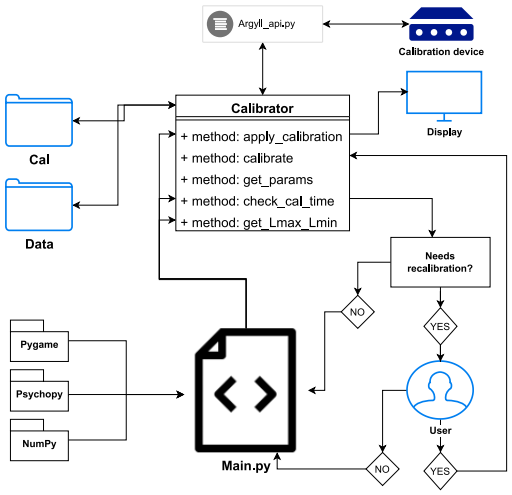


Figure 5.1: Calibration life-cycle diagram

Main.py file serves as the controller of the whole procedure. It handles interaction with the user and runs the application loop. All of the program data are stored in two folders:

1. Cal - stores all the calibration profiles
2. Data - store all the other data, like the time of the previous calibration or the monitor transfer function parameters

The calibrator class handles all the calibration related tasks. Mainly it implements the calibration procedure. For communication with the calibration device we use *Argyllcms*. *Argyllcms* is a command line tool which implements various calibrations and allows us to communicate with a wide variety of calibration devices. Our program was only tested with Xrites i1 Display Pro so we can't guarantee it will work with other devices, though it should if the device is supported by *Argyllcms*. The *argyll_api.py* file implements the API for communication with the *argyll* command line tool.

Calibration process

Calibration process involves two steps.

1. White point brightness calibration
2. Display transfer function estimation

Step 1 is a part where we set brightness of white to $120cd/m^2$. This is achieved either manually or automatically. Most monitors allow some form of communication with the PC (or microcontroller). The most common protocol is DDC which utilizes I2C communication. Rarely it can also use USB protocol. The program will first check if auto calibration is available. If yes then we auto adjust and move to the next step. If no manual adjustment is prompted and the user is asked to set the brightness accordingly.

Step 2 measures luminance of 33 gray levels which are then used to fit the transfer function in the form of:

$$L = ax^{\gamma} \tag{5.1}$$

Where:
a and γ are the parameters to be fitted
x is the gray level (values between 0 - 1786)

Following [11] we use the bit stealing technique to expand the range of possible luminance levels thus getting the values in the range of 0-1786. These values each correspond to rgb triplet in a LUT constructed according to [11]. In short the assume that color wise if we change one of the rgb channels by one the observer will no notice the color change. Like this for each grey level we can get 7 luminance levels around it. 255 cannot be combined with anything so we get $n = 7 * 255 + 1 = 1786$ the one is for (255, 255, 255) triplet. We use non-linear least squares to construct the function to be optimized. We use Levenberg-Marquardt method [24] to optimize this function.

Experiment life-cycle

The experiments life-cycle 3 is displayed in figure 5.2

After the experiment is commenced contrast and sensitivity (the stimuli parameters) are provided by the *qCSF.py* file. This file implements the inference algorithm. *Main.py* then requests the generation of the stimuli. Function called *grating_stimuli* then creates a stimuli accordingly. This stimuli is then sent back to *main.py* which then proceeds to show the stimuli to the user. User then provides a response. This response is provided to *qCSF.py* which calculates the next stimuli parameters and sends them to *main.py* thus beginning the next cycle.

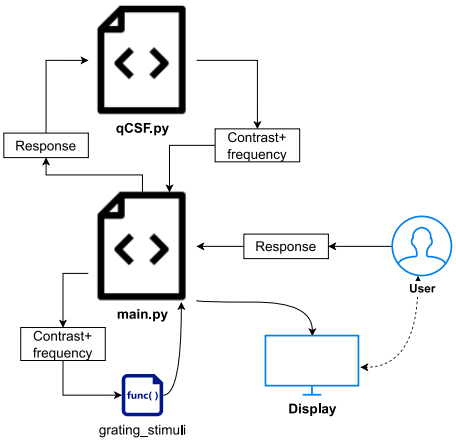


Figure 5.2: Experiment life-cycle diagram

5.2 Hardware setup

Program runs on Raspberry Pi 4 model B with 2GB RAM. External monitor is required. You should use at least a full HD monitor the maximum brightness should be $120cd/m^2$ minimum. The test is operated from the keyboard no mouse needed but to navigate the Raspbian OS mouse is recommended.

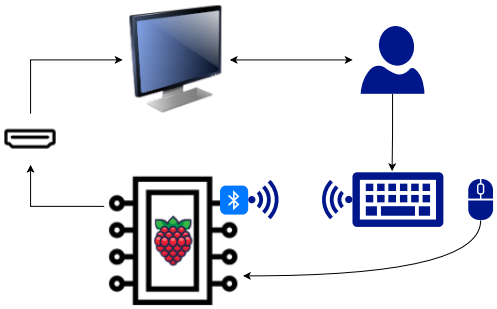


Figure 5.3: Hardware setup diagram

Why Raspberry Pi?

We chose this device because its small, portable, has all the connectivity we need (USB, HDMI) and can run linux base OS. Manufacturer provides their own operational system called Rasbian (based on aarch-linux). We use this OS mainly because other OS require higher RAM. This would be the main drawback of the lower RAM model since for example Ubuntu OS would be easier to navigate. The computational power is sufficient and we did no run into any memory problems with 2GB RAM.

Monitor connectivity

As far as hardware goes this is the main thing that can cause trouble. If you use monitor with HDMI there should not be a problem. But if you use older monitor with VGA you might encounter two problems:

1. Rpi is unable to supply the power needed for HDMI to VGA conversion. You may solve this by using an adapter which is directly powered from independent source.
2. Using VGA will most likely lead to problems with rpi->monitor communication. These problems are most likely caused by the adapter not correctly connecting I2C wires. Alternatively, I2C struggles with reliability when longer cables are used. If any of these problems occur it will disable automatic brightness calibration.

5.3 Calibration device validation

To asses suitability of Xrite i1 Display Pro[1] device for the calibration process, we used Konica Minolta CS-2000 spectroradiometer[2] as a reference device. CS-2000 allows measurement of the whole spectrum. The other values are then calculated from the spectrum. We are only interested in the luminance (Y part in the XYZ space).

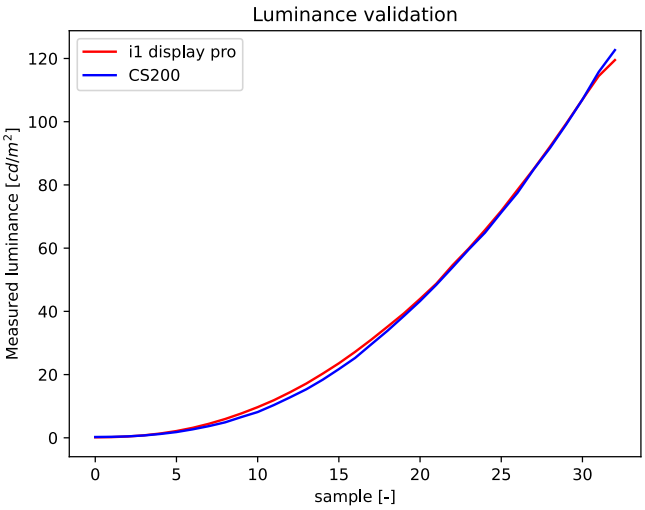


Figure 5.4: Measured luminances with two devices

Figure 5.4 plots interpolated luminances measured by the two mentioned devices.

[1] https://www.xrite.com/categories/calibration-profiling/i1display-pro
[2] https://sensing.konicaminolta.us/us/products/cs-2000-spectroradiometer/, luminance accuracy=±2%

Error analysis

The RMSE calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N}(L_{ref_i} - L_{m_i})^2}{33}} = 1.16 cd/m^2 \tag{5.2}$$

Where:
L_{ref_i} is the luminance measured by the CS-2000 (reference)
L_{m_i} is the luminance measured by the i1 Display Pro device
33 is the number of measured points

This value seems rather high but since we only use contrast which we define as a percenatage of the display luminance span (around 120 cd/m^2) we reason about the errors effects like this:

As the graph suggests the error doesn't change very fast so for two close luminances the difference between the errors isn't big. Since contrast depends only on the difference between the luminances its error does too. Considering this we can say that for two close luminances (which is the case most of the time) the error gets canceled.

Also we mostly care about luminances near 60 cd/m^2 where we can see the error is much smaller. The RMSE between 45-70 cd/m^2 is 0.59 cd/m^2.

Example:

Consider two luminances [3] $L_1 = 48.755 cd/m^2$ and $L_2 = 71.794 cd/m^2$ and maximum luminance span of $119.5 cd/m^2$. Then contrast c equals:

$$c = \frac{L_1 - L_2}{119.5} = 0.19 \tag{5.3}$$

The error on these luminances with respect to reference is:

$$\epsilon_1 = |L_1 - L_{ref1}| = |71.794 - 71.3| = 0.494 cd/m^2 \tag{5.4}$$

$$\epsilon_2 = |L_2 - L_{ref2}| = |48.755 - 48.4| = 0.355 cd/m^2 \tag{5.5}$$

Error on contrast ϵ_c then equals to:

$$\epsilon_c = \frac{\epsilon_1 - \epsilon_2}{119.5} = 10^{-3} \tag{5.6}$$

At contrast c a change of e_c will not be detected at all by the observer thus proving the omittability of the error.

5.3.1 Validation conclusion

We render the device sufficient for our use case on the grounds of the following arguments.

[3] Chosen from the measured levels

1. For low contrasts (which we mostly use) the introduced error is minimal.
2. The errors introduced by uncontrollable sources such as human exhaustion or the method itself are much higher.

Chapter 6

Results of psychophysical testing

To demonstrate the devices portability we conducted measurements with two different monitors (HP L2151ws and Eizo CG242W) placed in two different localities (FEE CTU and my home). In total three observers participated. I will call them S1, S2 and S3 for clarity. Unfortunately only S1 and S2 were able to undergo the experiment in both locations.
Note that we are not trying to statistically measure the population CSFs thus the low number of participants. The goal is just to demonstrate the portability of the device.

6.1 Experiment setups

All the participants were presented a 4-AFC test with qCSF evaluation and the stimuli described in 4.3. In the next table we summarize setups for each of the participants

	S1	S2	S3
Subject distance [cm]	150	150	150
Lighting conditions	light	dark	light
Number of trials	100	100	100
Corrected vision	No	Yes	Yes
Age	21	51	20

Table 6.1: Table of experiment setup for each participant

Table 6.2 summarizes parameters of the two used monitors. Looking at 6.2 we see that Eizo is the superior monitor. Especially in bit depth. As will the results show this will not play a very significant role.

	Refresh rate [Hz]	Max luminance[cd/m^2]	Bit depth [bit]	Resolution	Diagonal [in]
Eizo CG242W	60	270	12	1920 × 1200	24
HP L2151ws	60	250	8	1920 × 1080	21.5

Table 6.2: Table of experiment setup for each participant

We should also note that the HP monitor was old and the maximum luminance it could provide was around $190 - 200cd/m^2$. In the table 6.2 there are values provided by the manufacturers.

6.2 Subject one (S1) results

S1 was tested 4 times in total (3 times on Eizo monitor once on HP). The typical measurement time is between 5-15min based on the experience of the observer. Figure 6.1a shows all the functions measured, 6.1b the mean CSF ± standard deviation marked by the red area. The standard error for frequency f calculated as:

$$\epsilon = \frac{\sigma_f}{2} \tag{6.1}$$

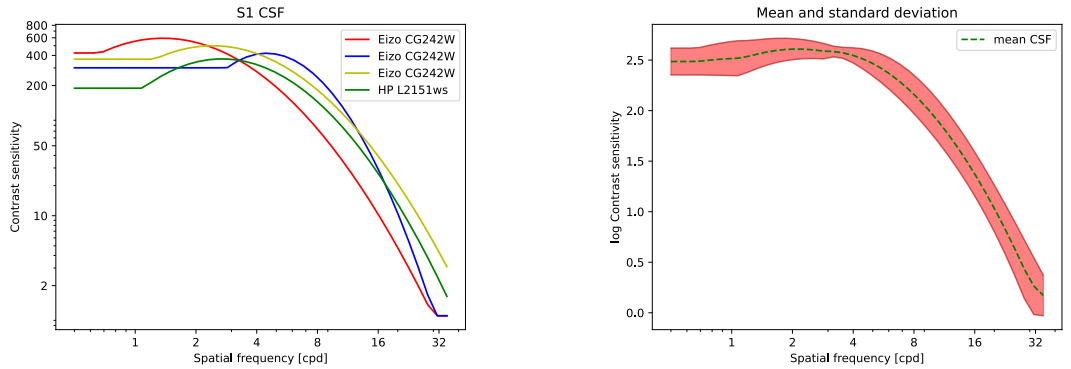
To get information about the whole fit in one number we took the mean of the standard errors calculated for each discrete frequency f.

$$\epsilon = 0.730, \ \epsilon_{log} = 0.0802 \tag{6.2}$$

Where ϵ_{log} is in log units. To asses the shape similarity we can use Pearson correlation. This metric is widely used in psychometrics but do bare in mind it is a measure of linear relationship. So it will be 1 if our estimates have the same shape but differ in scale. We calculate the Pearson correlation between each possible combination of measured function and take their mean again.

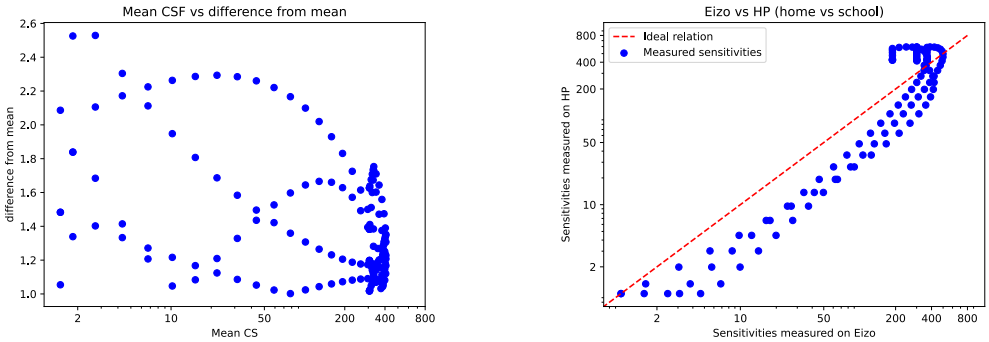
$$P_{corr} = 0.976 \tag{6.3}$$

We mentioned Pearson correlation has some shortages so now let us take a look on two graphical evaluation tools.
In figure 6.2a we observe how the sensitivities differ from the mean. On x axis we plot the mean sensitivity and on y axis the corresponding differences from the mean. This allows us to evaluate where the fits agree (clustered parts) and where they differ more (spread parts). For example near the higher values of sensitivity we see a big cluster. This cluster corresponds to the truncation at lower frequencies where the CSF's have more or less similar and constant (thus the cluster) values.



(a) : Mesured CSF's **(b) :** Mean and SD

Figure 6.1: (a) CSF's measured on S1 (b) Mean CSF ± standard deviation.



(a) : Differences from mean **(b) :** CSF home vs FEE

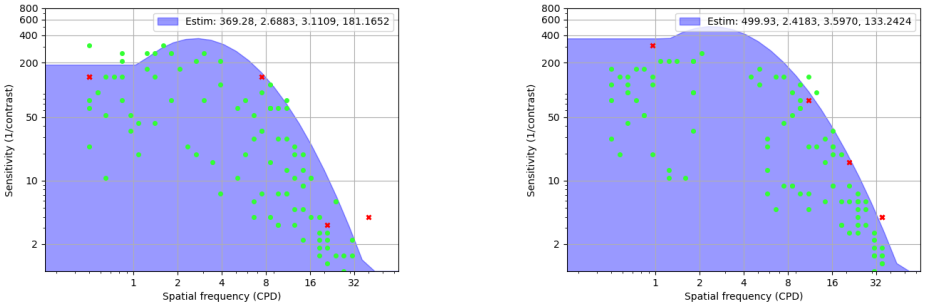
Figure 6.2: (a) Plots absolute difference from mean CSF for each measured CSF (b) Is a comparison between sensitivities measured at FEE and in my home.

Figure 6.2b shows how sensitivities measured at FEE correspond to the ones measured in my home. Ideally they would lay on a line defined by $y = x$ relation. This is the dashed red line.

6.2.1 Algorithm impact on the results

Figure 6.2b would suggest the location has indeed impact on the measurement. But partially this could be explained by closely observing the results of the test. In figure 6.3 we can see two of the four measured CSFs, one measured at home one at FEE. If we take a look at the CSF obtained at my home near the left upper side we can see a red cross. Cross indicates incorrect answer to the stimuli defined by its position. This answer was a miss-click from S1. Since it was in a later trial the method didn't have "time" to recover from it and it skewed

the results a little bit.
The correct answer above the cross also proves S1 was able to see higher contrasts than this one.



(a) : Home **(b) :** FEE

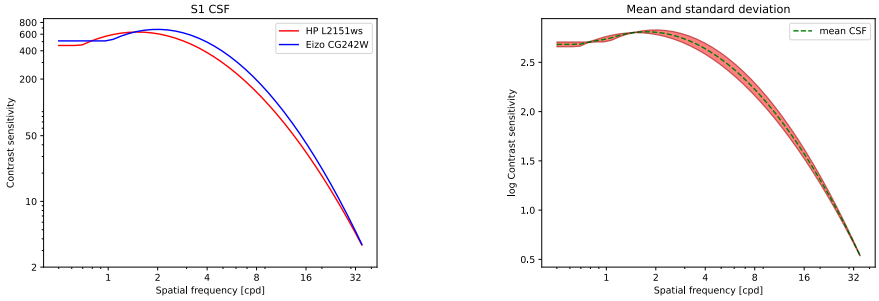
Figure 6.3: Shows comparison of experiments conducted at FEE and at my home.

6.3 Subject two (S2) results

S2 completed two runs one on HP L2151ws (home) and one on Eizo CG242W (FEE). Calculating error from such few samples is redundant so we replace it with the mean absolute difference between the two measurements as it seems more appropriate:

$$\delta = \frac{\sum_f |CSF_1(f) - CSF_2(f)|}{N} = 46.244 \tag{6.4}$$

Where N is the number of discrete spatial frequencies.
Figure 6.4 shows the measured CSFs along with the mean and standard deviation plotted in red as in 6.1

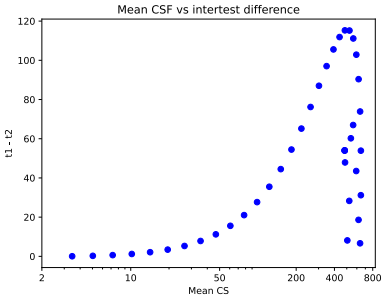


(a) : Measured CSFs **(b) :** FEE

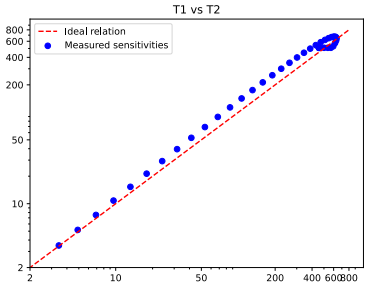
Figure 6.4: (a) Estimated CSF (b) Mean CSF ± standard deviation.

Since we only have two measurements we use a different metric. In figure 6.5a we plot absolute difference between the two measured CSFs against their mean CSF. This metric

shows us the span of the sensitivities around their mean value. We also use the same way to asses the relation between the two testing sites in figure 6.5b. Since there were no errors induced by uncontrollable sources the results are much closer to the theoretical perfect line.



(a) : Difference against mean CSF



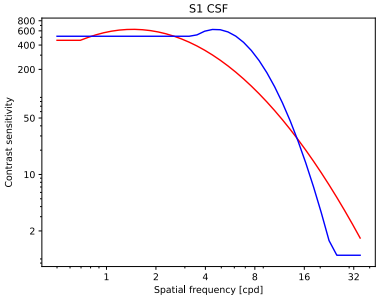
(b) : CSF home vs FEE

Figure 6.5: (a) Inter-test difference vs mean sensitivity (b) Sensitivities measured at FEE vs those at home.

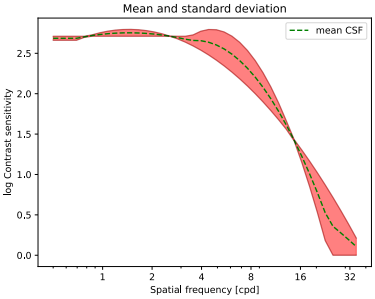
6.4 Subject three (S3) results

S3 participated again in two testing sessions, both of them with HP monitor (my home). We again calculate the mean absolute difference. In figure 6.6 we again see the measured CSFs along with their mean ± SD plot.

$$\delta = 91.55 \tag{6.5}$$



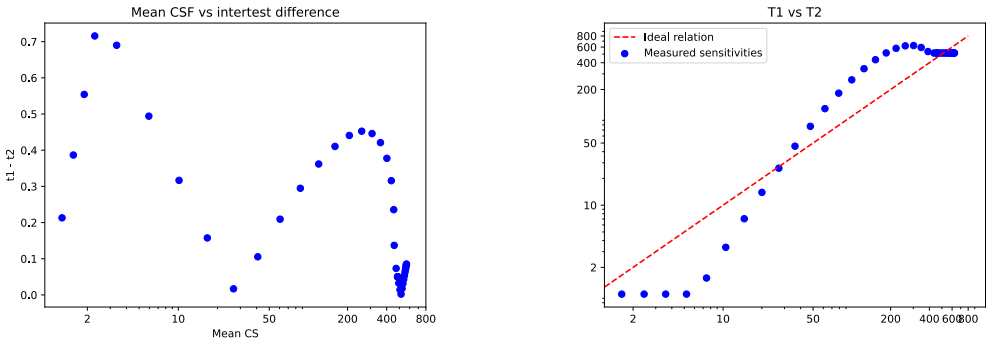
(a) : Measured CSFs



(b) : FEE

Figure 6.6: (a) Estimated CSF (b) Mean CSF ± standard deviation.

In figure 6.7a the difference is plotted in log units for easier interpretation. We notice two distinct peaks, one on lower sensitivities and one around 200-400. The first peak corresponds to the mismatch on higher frequencies 6.6a. The second one corresponds to the mismatch in peak frequency parameter.

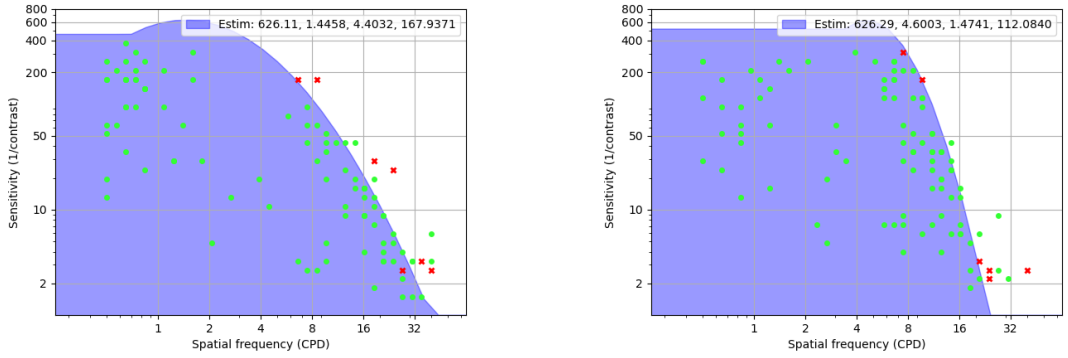


(a) : Difference against mean CSF **(b) :** CSF home vs FEE

Figure 6.7: (a) Inter-test difference vs mean sensitivity (b) Sensitivities measured in the successive runs plotted against each other.

6.4.1 S3 CSF estimate mismatch breakdown

Let us investigate the mismatch at higher frequencies.
In figure 6.8b we see three incorrect answers (red crosses) at frequencies around 20 cpd. The participant claims one of these was a mistake. Both the other stimuli were introduced in latter trials (89 and 54). About the 89th trial stimuli we can speculate that the subject was exhausted since both of the measurements were done consecutively, run 2 being the second one. Effectively it was 189th trial for the subject. This further confirms the hypothesis that human lapse influence rises with trial number.



(a) : Run 1 **(b) :** Run 2

Figure 6.8: S3 test course plots.

6.5 Results summary

The main goal was to demonstrate the flexibility of the device. Considering the results we can conclude that the environment had some influence on the measurement but the most significant factor at play seemed to be the subjects performance oscillations. Interestingly S3's results were the most inconsistent. This was mainly due to S3 lapse and exhaustion but this is speculative the methods reliability came at play too.
We proved that the main struggle when dealing with psychometric tests is indeed the human intervention which we try to control as much as we can.

We will not be doing any statistically based inferences from the performed experiments since the data are not sufficient. The method's performance was already evaluated in [2].

Chapter 7

Conclusion

The main goal of the thesis was to provide a summary of basic characteristics of HVS and develop a portable measurement device capable of measuring one of them. The emphasis was given mainly on device's accessibility in terms of its price and the components.

We choose to measure CSF because it seems to be a very comprehensive characteristic of the HVS but there are very few devices which allow us to measure it. Compared to visual acuity, research focused on CS is rather sparse. In hope of deepening collective knowledge about CSF we chose to provide a device which will make its measurement more accessible both economically and positionally without a significant loss in accuracy.

Compared to the commercially available instruments our devices proves to be more economical and portable. The components are accessible at typical stores. The biggest drawbacks against commercially available devices would prove to be its focus on only one characteristic. Mostly this device would not be suitable for medical applications since they often do not require a full CSF measurement. Also in medicine it is more common to measure multiple different characteristics of HVS and having a different device for each of them is unnecessary.

All in all the device is suitable for informed users. It provides better user experience then other lab-like implementations but worse than commercial products. The main usage would in lab environment where individuals vision is examined more thoroughly. For example when researching the influence of different diseases on contrast sensitivity you might want to perform more comprehensive test like our. Once you gather the results you can add a simplified test to the commercial devices which will focus on one part of the CSF (for example high frequencies).

We also provided a more thorough explanation of the qCSF algorithm than the original papers provide. To prove the portability of the device we performed experiments with two different monitors in two different environments. Although there were noticeable differences

between the tests performed in individual environments, the results suggest these differences are minor. Be aware that the results are only demonstrative. There was not enough measurements performed for us to make any conclusions based on statistical proof.

Most significantly we learnt that the influence of the tested subject on the test results can inhibit the ones caused by the distinct testing conditions. We demonstrated how subjects lapse in later trial can skew the tested results.

The main downsides to our device would be the requirements on user experience with computers and electronics. Even though operating Raspberry Pi is simple the installation progress might not feel very straightforward for inexperienced user. Further the applications graphical side is rather functional than pretty. The data logs gathered by the application provide all the info you need but their form is more oriented on processing in scripting environments like MATLAB or python. For users without experience in any of these environments migrating the data to Excel might be more strenuous then it needs to.

As far as accuracy goes we implement 4-AFC task which should perform better in theory than the original papers [2] 2-AFC test but there exists an article using a 10-AFC task[15] further improving performance. The main concern accuracy wise is the change of environment. We mainly focus on mitigating these problems but in comparison to tests performed with highly controlled conditions and on professional displays our device will probably fall short.

For future work this thesis can provide a good foundation both theoretical and practical for development of other HVS characteristics tests. The expansion could be to the color space. Instead of using monochromatic contrast we could use red-green or yellow-blue gratings. Further the CSF can be grown into the contrast sensitivity surface which introduces temporal frequency domain. The surface parametrization could look something like in figure 7.1.

At last this thesis provides an overview about contrast sensitivity and can be used as a learning material for the ones interested. In addition it gives the option to conduct a lab experiment and not just learn the theory.

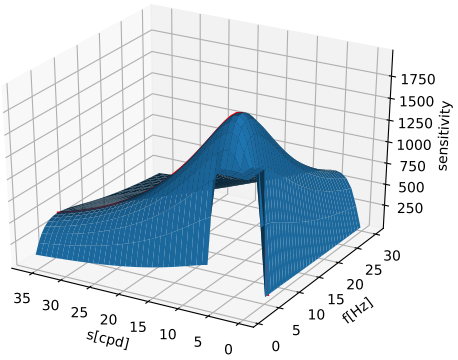


Figure 7.1: Parametrized contrast sensitivity surface.

Bibliography

[1] Snellen chart, Apr 2022. URL: https://en.wikipedia.org/wiki/Snellen_chart.

[2] Lesmes L. A. and Lu Z.-L. et al. Bayesian adaptive estimation of the contrast sensitivity function: The quick CSF method. *Journal of Vision*, 10(3):17–17, 03 2010. arXiv:https://arvojournals.org/arvo/content_public/journal/jov/933044/jov-10-3-17.pdf, doi:10.1167/10.3.17.

[3] Rocío Alcalá-Quintana and Miguel García-pérez. Stopping rules in bayesian adaptive threshold estimation. *Spatial Vision*, 18(3):347 – 374, 2005. URL: https://brill.com/view/journals/sv/18/3/article-p347_5.xml, doi:https://doi.org/10.1163/1568568054089375.

[4] Bulens C., Meerwaldt J. D., van der Wildt G. J., and Keemink C. J. Contrast sensitivity in parkinson's disease. *Neurology*, 36(8):1121–1121, 1986. URL: https://n.neurology.org/content/36/8/1121, arXiv:https://n.neurology.org/content/36/8/1121.full.pdf, doi:10.1212/WNL.36.8.1121.

[5] Azzam D. and Ronquillo Y. Snellen chart. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), January 2022.

[6] Canare D., Ni R., and Lu T. An open-source implementation of the quick csf method. *Journal of Vision*, 19:86b, 09 2019. doi:10.1167/19.10.86b.

[7] Amparo M. Díez-Ajenjo and Capilla P. Spatio-temporal contrast sensitivity in the cardinal directions of the colour space. a review. *Journal of Optometry*, 3(1):2–19, 2010. URL: https://www.sciencedirect.com/science/article/pii/S1888429610700029, doi:https://doi.org/10.3921/joptom.2010.2.

[8] Ghasemi M. et al. Comparison of visual status of iranian military and commercial drivers. *Iranian Red Crescent Medical Journal*, 17, 04 2015. doi:10.5812/ircmj.17(4)2015.19751.

[9] Shannon L. Risacher et al. Visual contrast sensitivity in alzheimer's disease, mild cognitive impairment, and older adults with cognitive complaints. *Neurobiology of Aging*, 34(4):1133–1144, 2013. URL: https://www.sciencedirect.com/science/article/pii/S0197458012004290, doi:https://doi.org/10.1016/j.neurobiolaging.2012.08.007.

[10] Sim Dawn A. et al. The Effects of Macular Ischemia on Visual Acuity in Diabetic Retinopathy. *Investigative Ophthalmology & Visual Science*, 54(3):2353–2360, 03 2013. arXiv:https://arvojournals.org/arvo/content_public/journal/iovs/933253/i1552-5783-54-3-2353.pdf, doi:10.1167/iovs.12-11103.

[11] Tyler Ch. et. al. Bit stealing: How to get 1786 or more gray levels from an 8-bit color monitor. volume 1666, 08 1992. doi:10.1117/12.135981.

[12] Wichmann F. and Hill N. The psychometric function: I. fitting, sampling, and goodness of fit. *Perception & Psychophysics*, 63:1290–1313, 11 2001. doi:10.3758/BF03194544.

[13] Pelli D. G., J. G. Robson, and A. J. Wilkins J. The design of a new letter chart for measuring contrast sensitivity. *Clinical Vision Sciences*, pages 187–199, 1988.

[14] Arden GB and Jacobson JJ. A simple grating test for contrast sensitivity: preliminary results indicate value in screening for glaucoma. *Investigative ophthalmology & visual science*, 17(1):23–32, 1978.

[15] Zheng H. and Wang et al. Measuring the contrast sensitivity function using the qcsf method with 10 digits. *Translational Vision Science & Technology*, 7:9, 11 2018. doi:10.1167/tvst.7.6.9.

[16] Kai-Qi Huang, Qiao Wang, and Zhen-Yang Wu. Natural color image enhancement and evaluation algorithm based on human visual system. *Computer Vision and Image Understanding*, 103(1):52–63, 2006. URL: https://www.sciencedirect.com/science/article/pii/S1077314206000233, doi:https://doi.org/10.1016/j.cviu.2006.02.007.

[17] Hontsch I. and Karam L.J. Locally adaptive perceptual image coding. *IEEE Transactions on Image Processing*, 9(9):1472–1483, 2000. doi:10.1109/83.862622.

[18] Sethi I.K. Entropy nets: from decision trees to neural networks. *Proceedings of the IEEE*, 78(10):1605–1613, 1990. doi:10.1109/5.58346.

[19] Gilchrist J., Jerwood D., and H Ismaiel. Comparing and unifying slope estimates across psychometric function models. *Perception & psychophysics*, 67:1289–303, 11 2005. doi:10.3758/BF03193560.

[20] Janne V. Kujala and Tuomas J. Lukka. Bayesian adaptive estimation: The next dimension. *Journal of Mathematical Psychology*, 50(4):369–389, 2006. URL: https://www.sciencedirect.com/science/article/pii/S0022249605001306, doi:https://doi.org/10.1016/j.jmp.2005.12.005.

[21] Shiow-Wen L. and Cheng-Jen Ch. Myopia and contrast sensitivity function. *Current Eye Research*, 22(2):81–84, 2001. arXiv:https://doi.org/10.1076/ceyr.22.2.81.5530, doi:10.1076/ceyr.22.2.81.5530.

[22] Bach M. The freiburg visual acuity test–automatic measurement of visual acuity. *Optometry and vision science : official publication of the American Academy of Optometry*, 73 1:49–53, 1996.

[23] Rohaly A. M. and Owsley C. Modeling the contrast-sensitivity functions of older adults. *J. Opt. Soc. Am. A*, 10(7):1591–1599, Jul 1993. URL: http://opg.optica.org/josaa/abstract.cfm?URI=josaa-10-7-1591, doi:10.1364/JOSAA.10.001591.

[24] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

[25] Karen A. Panetta, Eric J. Wharton, and Sos S. Agaian. Human visual system-based image enhancement and logarithmic contrast measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(1):174–188, 2008. doi:10.1109/TSMCB.2007.909440.

[26] Wang Q. R. and Suen Ch. Y. Analysis and design of a decision tree based on entropy reduction and its application to large character set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4):406–417, 1984. doi:10.1109/TPAMI.1984.4767546.

[27] Jesse Richman and Lorenzana et al. Importance of Visual Acuity and Contrast Sensitivity in Patients With Glaucoma. *ARCHIVES OF OPHTHALMOLOGY*, 128(12):1576–1582, DEC 2010. doi:{10.1001/archophthalmol.2010.275}.

[28] John Robson. Spatial and temporal contrast-sensitivity functions of the visual system. *Journal of The Optical Society of America*, 56, 08 1966. doi:10.1364/JOSA.56.001141.

[29] Kontsevich L. L.and Tyler Ch. W. Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, 39(16):2729–2737, 1999. URL: https://www.sciencedirect.com/science/article/pii/S0042698998002855, doi:https://doi.org/10.1016/S0042-6989(98)00285-5.

[30] Andrew Watson and Denis Pelli. Quest: A bayesian adaptive psychometric method. *Perception & psychophysics*, 33:113–20, 03 1983. doi:10.3758/BF03202828.

[31] Andrew B. Watson. QUEST+: A general multidimensional Bayesian adaptive psychometric method. *Journal of Vision*, 17(3):10–10, 04 2017. arXiv:https://arvojournals.org/arvo/content_public/journal/jov/936104/i1534-7362-17-3-10.pdf, doi:10.1167/17.3.10.

[32] Billy R. Wooten, Lisa M. Renzi, Robert Moore, and Billy R. Hammond. A practical method of measuring the human temporal contrast sensitivity function. *Biomed. Opt. Express*, 1(1):47–58, Aug 2010. URL: http://opg.optica.org/boe/abstract.cfm?URI=boe-1-1-47, doi:10.1364/BOE.1.000047.

Appendices

Appendix A

List of attachments

Source code

Provided source in zip attachment, also available on gitlab (https://gitlab.fel.cvut.cz/karlipe2/qcsf.git):

- **cal** - Folder with calibration related data used by the app. Required for correct app execution.
- **logs** - Folder which the app uses for saving the results. Required for correct result saving.
- **data** - Folder with other internal app data. Mostly stores states between distinct application runs. Required for correct execution.
- *exec.py* - this executes the main application cycle.
- *QuickCSF.py* - qCSF implementation from [6] altered by me to fit our app.
- *simulate.py* - Another script from [6]. Altered by me to generate nicer plots and prepare them for saving. Also provides simulation functions.
- *grating.py* - Script with function responsible for the stimuli generation.
- *argyll_api.py* - *ArgyllCMS* API implementation (communication with calibration device).
- *Calibrator.py* - Implements the class which is responsible for calibration related tasks.
- *CalibrationApp.py* - Executable abstraction for *Calibrator* class.
- *helper_funcs.py* - Functions for unit conversion and file handling.

- *screens.py* - Some of pygame screens used by the app.
- *data_proc.py* - Fucntion for loading the results to Jupyter notebook for example
- *Data_processing.ipynb* - Jupyter notebook used to generate all the plots and calculations presented here.
- *setup.py* - Build script if you wish to build the executables yourself
- *dependency_install.sh* - Bash script which installs dependencies. Handy when executing from source.
- README - readme manual

Executables

Only on gitlab (https://gitlab.fel.cvut.cz/karlipe2/qcsf.git) there is a folder **qCSF_app** with executables:

- *main.sh* - Script to run the main application
- *CalibrationApp* - Executable to run the calibration app

Folder structure is the same as in the **src** folder.

Appendix B

CSF software installation

This attachment describes the procedure for software installation and provides solution to the most common problem of missing c libraries on Raspberry Pi.

Installation

There are two ways to run the program.

1. Run the python script directly
2. Use the compiled distribution

Mostly it is beneficial to use the distributed folder and run the program from there as you don't have to go through the dependencies installation. If for some reason the compiled distribution doesn't run you will have to run the script directly.
I created a script which you have to run as root in the command line. The following command will do

```
$ sudo ./dependecy_install.sh
```

It is probable that while running this script you will encounter include error. To solve this just find the package which contains the include and install it. For example you get error such as this:

```
#include <pulse/pulseaudio.h>
         |              ^~~~~~~~~~~~~~~~~~~~
  compilation terminated.
  error: command 'gcc' failed with exit status 1
```

You see the include fail here. To solve this we need to install the "libpulse-dev" package. This command will do:

```
$ sudo apt install libpulse-dev
```

[language=bash] To find the package with the missing include you can use the internet but apt-file might be a better option. Command like this will find our missing package:

```
apt-file search pulse/pulseaudio.h
```

Note that most of these packages should be installed by the script. But this will depend on the version of the OS which the Raspberry Pi has.

Running the program

In the first case just navigate to the program directory and run the command:

```
$ python3 main.py
```

In the second case in the compiled folder find the "main" executable and either run it from command line:

```
$ ./main.sh
```

or you can make a reference to this executable and place it on the desktop.

Appendix C

Measuring the CSF

I will explain how to measure the CSF and perform calibration using the provided software + hardware.

Running the calibration

First you should calibrate your display to ensure results reliability. You will need to run the CalibrationApp script. If you use my device there should already be an Icon on Desktop, just click it. If you use the executables you need to run it from the command line. Navigate to the directory which hold the CalibrationApp executable and run:

```
$ ./CalibrationApp
```

Alternatively to run it from source run:

```
$ python3 CalibrationApp.py
```

When you run the app you will be asked to connect and place the calibration probe on the screen. Once you do that the calibration can begin.
There are two parts to the calibration process.

1. White point brightness calibration
2. Monitor transfer function parameters estimation

First the app will check if autocalibration of 1 is available. After this you will be notified about the results. If manual calibration is required be sure to follow the instructions. Mainly adjust the brightness on the MONITOR not on keyboard. Once you are done start part 2.

Running the experiment

Again you can use the icon if available, else running this command in the executable directory will do. Navigate to the directory and run the following command.

```
$ ./main.sh
```

From source:

```
$ python3 main.py
```

First you will need to setup the experiment. The input fields are:

1. Experiment name - will be used as the file name if you wish to save the experiment (default: 'results_log')
2. Number of trials
3. Screen diagonal in inches
4. Display resolutions (width, height in pixels). Should already be filled as we can fetch the info from the computer so it only needs to be checked
5. Subject distance - distance of the tested person from the display

After you fill everything press OK. You will be prompted with an instruction screen. You can see the instruction screen in figure C.1.

Advice to instructions

Please read the instructions displayed carefully. Don't be shy to use the focus screen (holding SPACE). If you feel like your eyes are burning or you can't see the stimuli try to hold SPACE for a bit to relax your eyes. Trust me this helps a lot. Also if you loose the center of the screen use it too, a white cross will fixate you back.

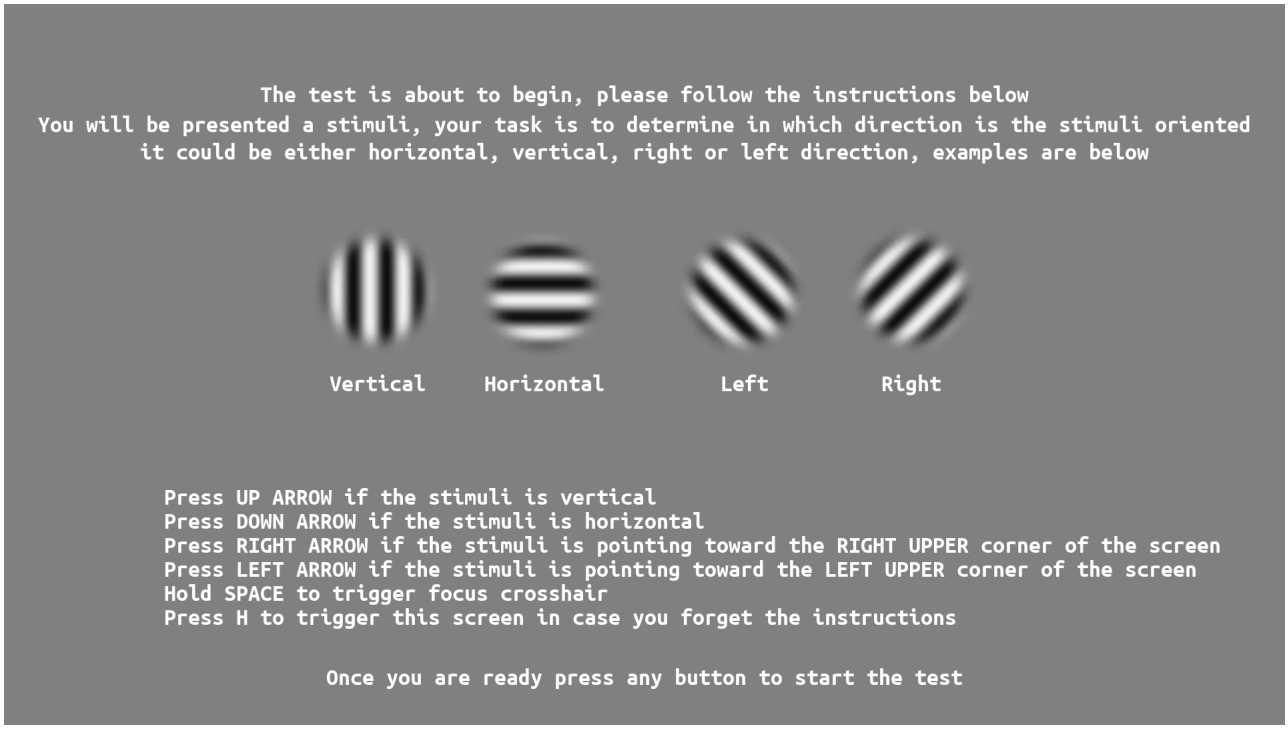


Figure C.1: The instructions screen

Extracting the measurement data

After the experiment finishes you will be prompted with a save screen. If you wish to save the results press Y otherwise N.
The program will save the data in a zip file. The name of this zipfile is determined by the test name you input before commencing the test.
The file will be stored in Logs folder which sits in the same directory as the executable/script you use to run the program.

Zip file structure

The parent zip contains three files:

1. Result_parameters.csv - includes the estimated parameters.
2. Experiment_log.csv - includes contrast, frequency and the subject's response (correct - 1, incorrect - 0) for each trial.
3. fig.png -is a picture of the estimated CSF. The dots mark the stimuli that was asked and their color represents the response (correct - green, incorrect - red) of the subject to the stimuli.

Understanding the results

The results are quite easy to understand. Fig.png will be your main concern. This plot shows you the estimated CSF with all of your answers as dots or crosses depending on your response. Also I provide some data processing functions for your convenience. These functions are in data_proc.py file. The important one is the read_params function. It takes the zip file name (string) as input and will output the CSF values. You can then use them to make your own plots.
The second function (mean_pearson_corr) takes a list of numpy arrays of CSF values (for example returned by read_params). It calculates Pearson correlation between every pair of provided CSFs and returns their mean.