WWU MÜNSTER

Eyetracking Workshop WS 18/19

Wanja Mössing

November 2018

- AlternativeTurorialbox
- Today we will use 1 Eyetracker!

- Create account at sr-support

Today

1. Short: How does the Eyelink 1k+ work?
2. How to do gaze control in PTB & Psychopy?
3. Preparing EEG/ET coregistration in your experiment
4. Storing behavioral data in your eyetrack
5. Proper calibration
6. Converting .edf files: stupid (Matlab) & smart (R)
7. Preprocessing .edf files in R

Eyelink

How does the EL1000+ work?

Huey (1908)

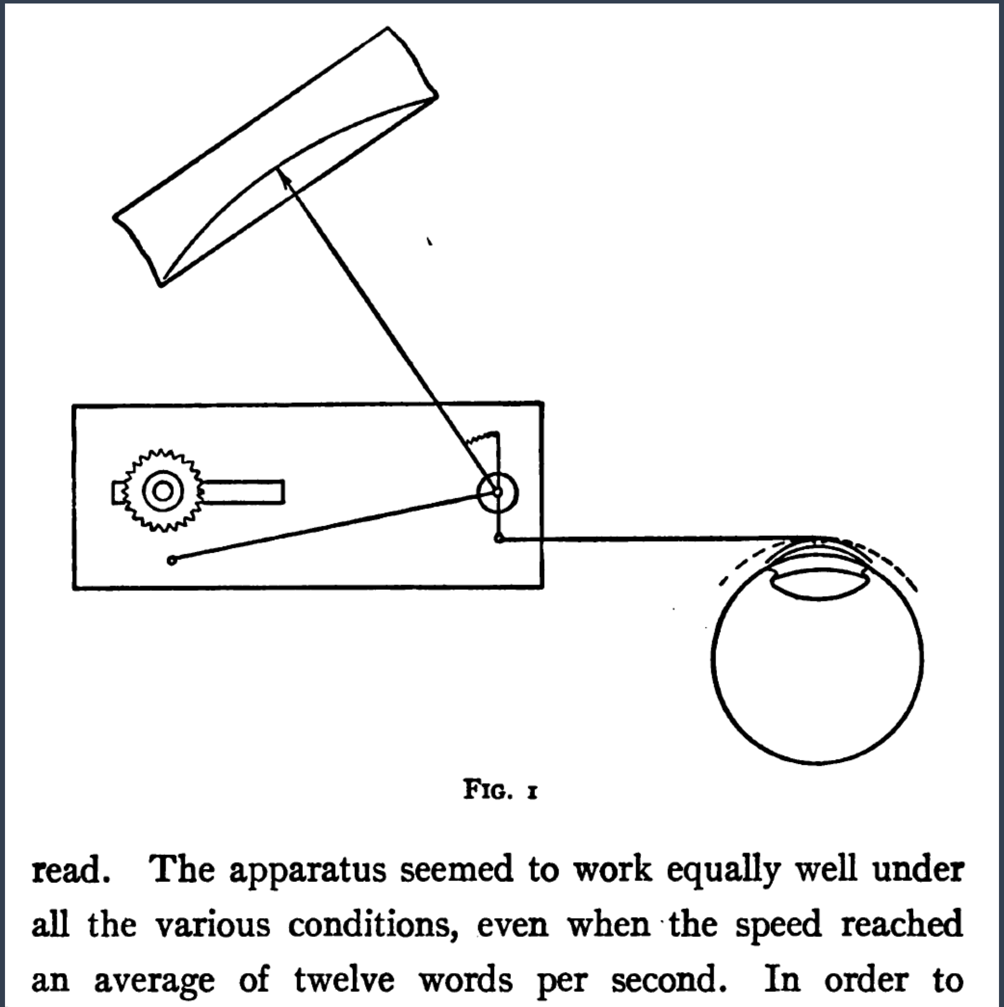


Fig. 1

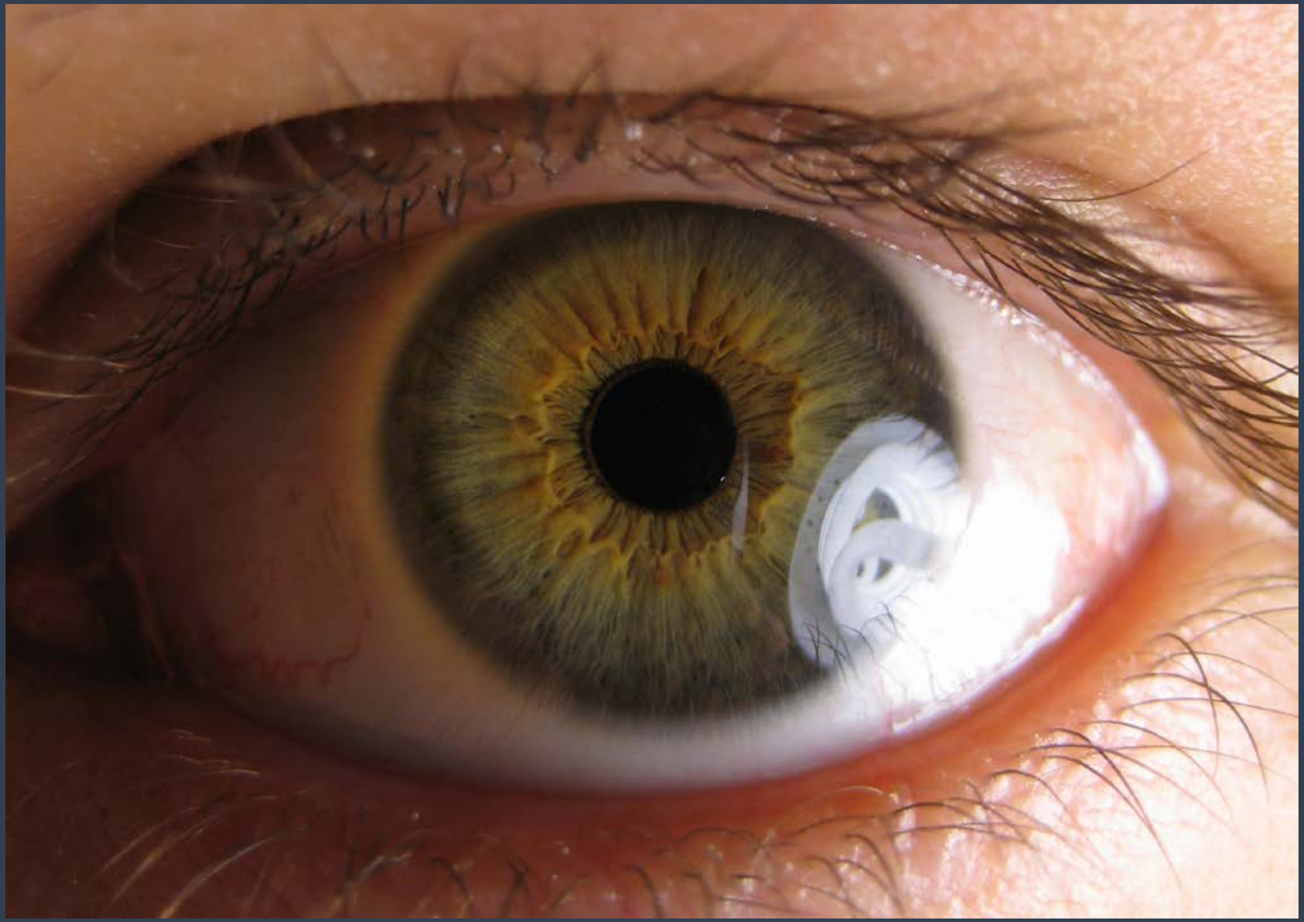
read. The apparatus seemed to work equally well under all the various conditions, even when the speed reached an average of twelve words per second. In order to

EL1k+

- Infrared based eyetracker
- Infrared light is emitted to the eye
- The lens records the eye from another angle
- EL-host-PC detects pupil and Purkinje image

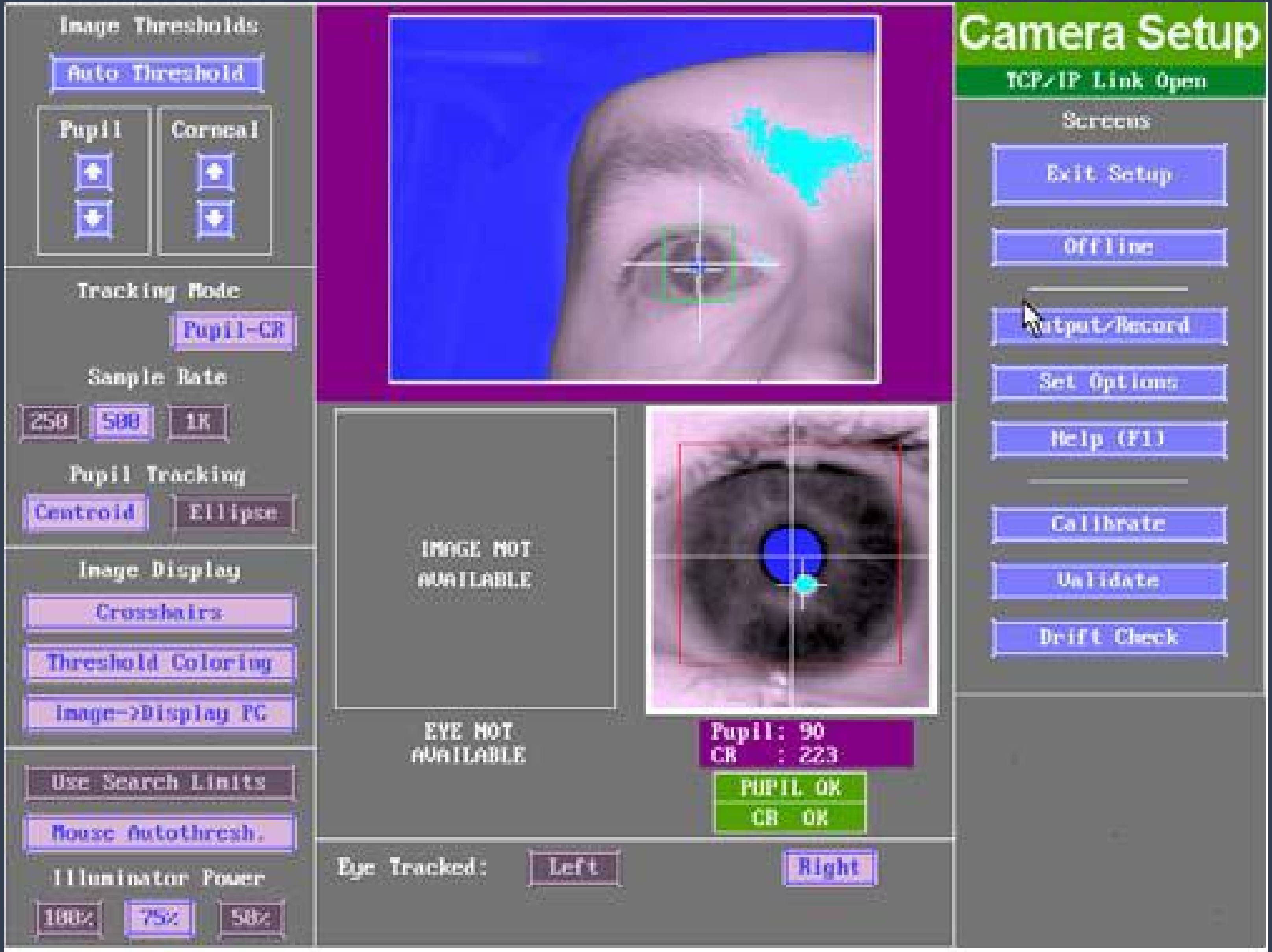
Purkinje

pupil



Purkinje images

Calibration



Camera Setup

TCP/IP Link Open

IMAGE NOT AVAILABLE

EYE NOT AVAILABLE

Pupil: 90
CR : 223

PUPIL OK
CR OK

calibration

- Calibration creates constellation-dependent coordinate system
- Validation takes similar measurements and checks how much the screen position differs from the system‘s prediction

Software structure

Matlab / python

Install this (to record)

- SR Developer‘s pack
- Psychtoolbox comes with Eyelink toolbox
- Psychopy comes with *too outdated* Eyelink module
  - See my readme on github for how to update (don‘t use pip!)
- My set of high-level Eyelink functions
  - github.com/wanjam/wm_utilities/Eyelink

Install this (to analyze)

- SR Developer‘s pack
- Matlab:
  - Coregister EEG/ET: Olaf Dimigen‘s <EYE-EEG>
    - Eeglab plugin manager is outdated. Use github: olafdimigen/eye-eeg
- R:
  - Parse edf files: Jason Hubbard‘s <edfr>
    - devtools::install_github(‘jashubbard/edfr‘)
  - Massage data: Hedderik van Rijn‘s <pR>, my <wmR>
    - devtools::install_github(‘wanjam/wm_utilities/helper-funs/wmR‘)
    - devtools::install_github(‘hedderik/pR‘)

SR-dev kit

- SR research provides a „developer‘s kit“
- Newest version always in forum
- Ubuntu: PPA (read instructions *very* careful)
- Windows/Mac: zip/dmg
- Core of everything (written in C++) → install!

- Have a look: sr-support.com

Software structure

EDF datafiles

SR’s developer’s kit

Jason hubbard “edfr”

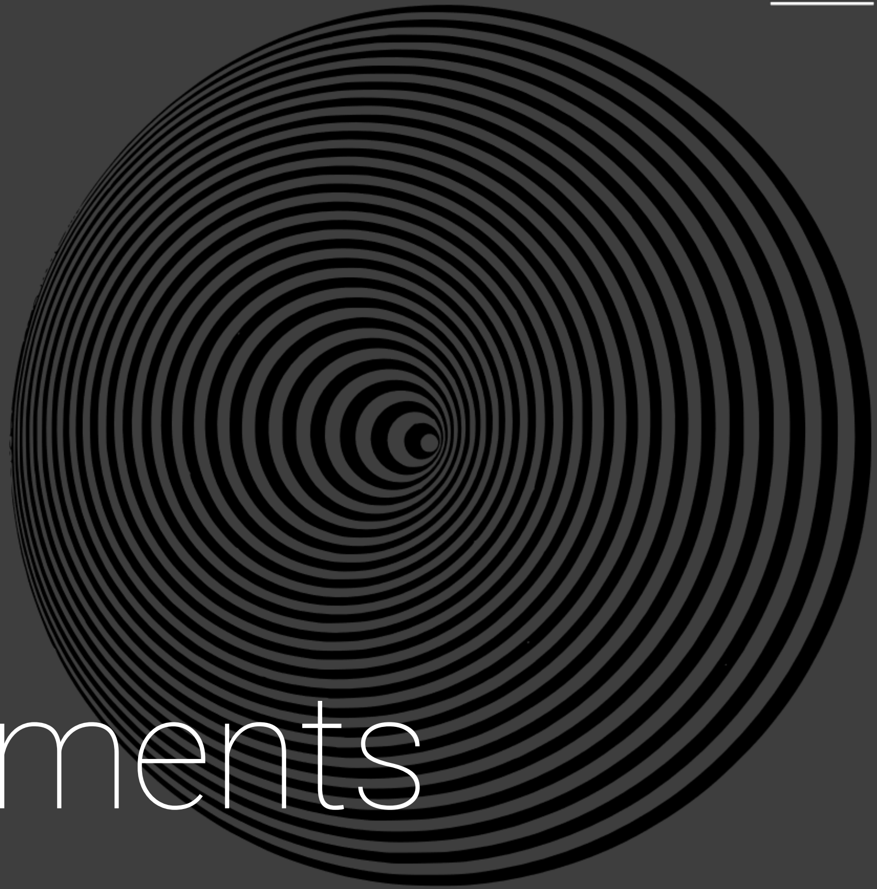
Analysis

wm_utilities/Eyelink

Stimulation PC

Psychtoolbox /Eyelink

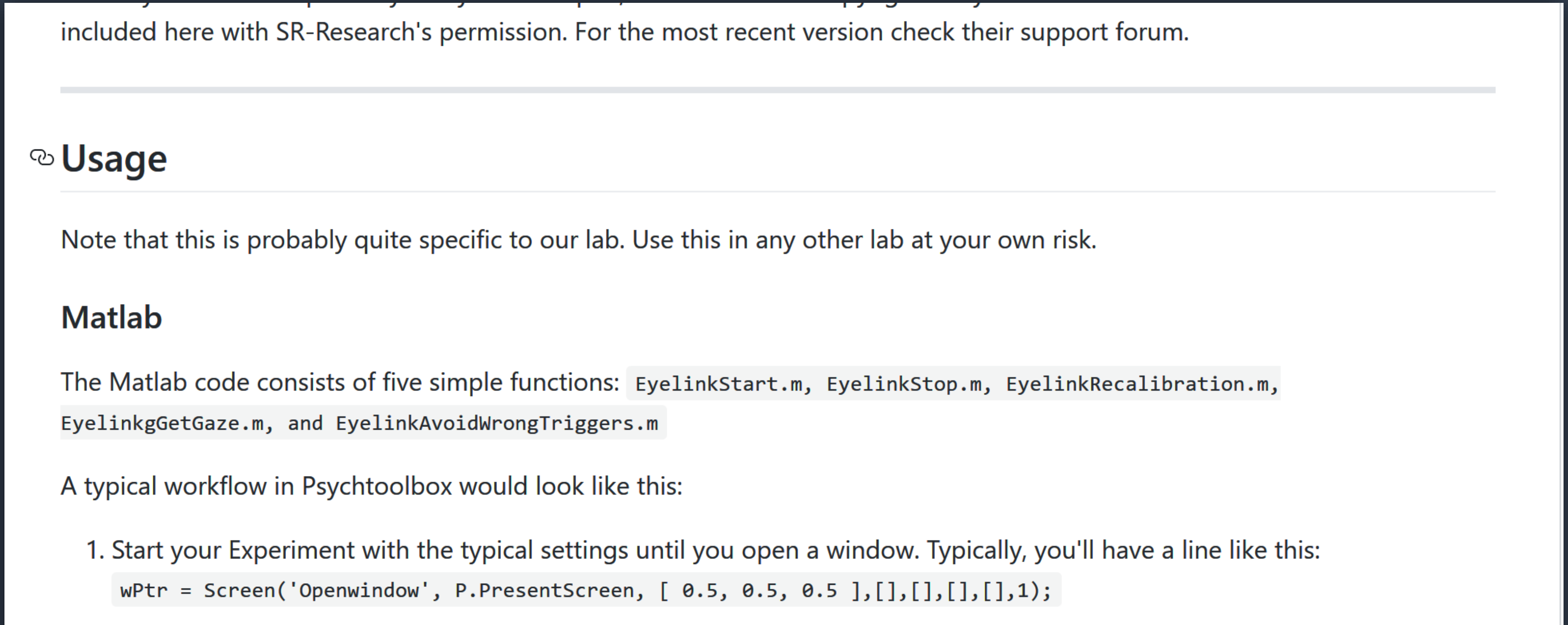
Gaze-control



Functions for Experiments

wm_ utilities

Readme online! https://github.com/wanjam/wm_utilities/Eyelink



included here with SR-Research's permission. For the most recent version check their support forum.

Usage

Note that this is probably quite specific to our lab. Use this in any other lab at your own risk.

Matlab

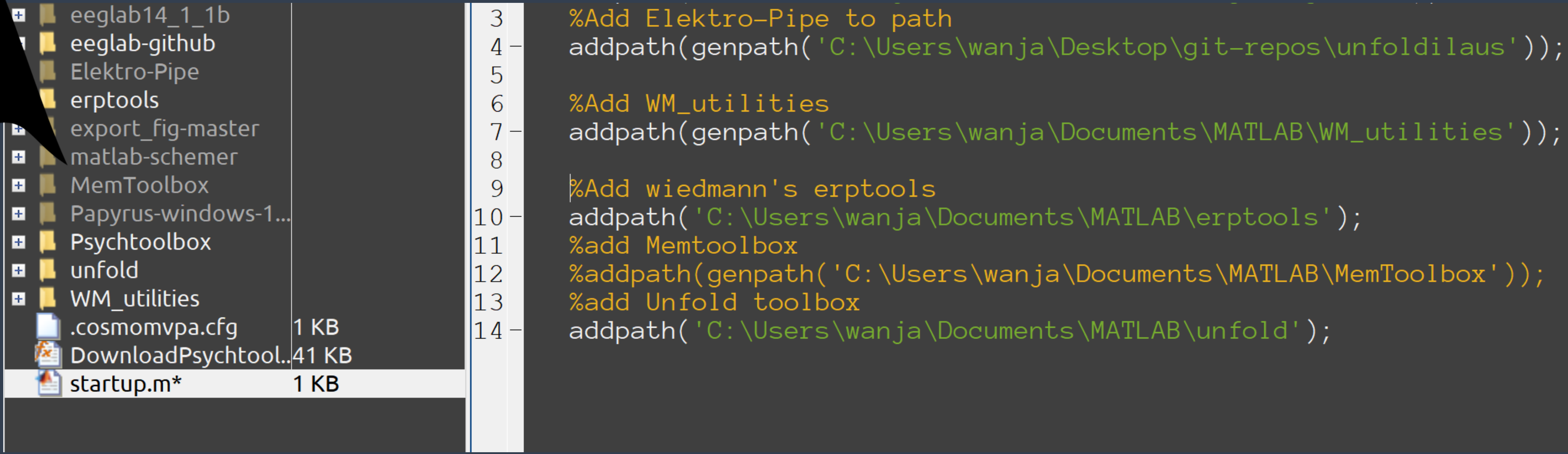
The Matlab code consists of five simple functions: `EyelinkStart.m, EyelinkStop.m, EyelinkRecalibration.m, EyelinkgGetGaze.m, and EyelinkAvoidWrongTriggers.m`

A typical workflow in Psychtoolbox would look like this:

1. Start your Experiment with the typical settings until you open a window. Typically, you'll have a line like this:
   ```
   wPtr = Screen('Openwindow', P.PresentScreen, [ 0.5, 0.5, 0.5 ],[],[],[],[],1);
   ```

wm_ utilities

I‘ll only talk about Matlab – Same functions for Python!



```
eeglab14_1_1b
eeglab-github
Elektro-Pipe
erptools
export_fig-master
matlab-schemer
MemToolbox
Papyrus-windows-1...
Psychtoolbox
unfold
WM_utilities
.cosmomvpa.cfg          1 KB
DownloadPsychtool..41 KB
startup.m*              1 KB
```

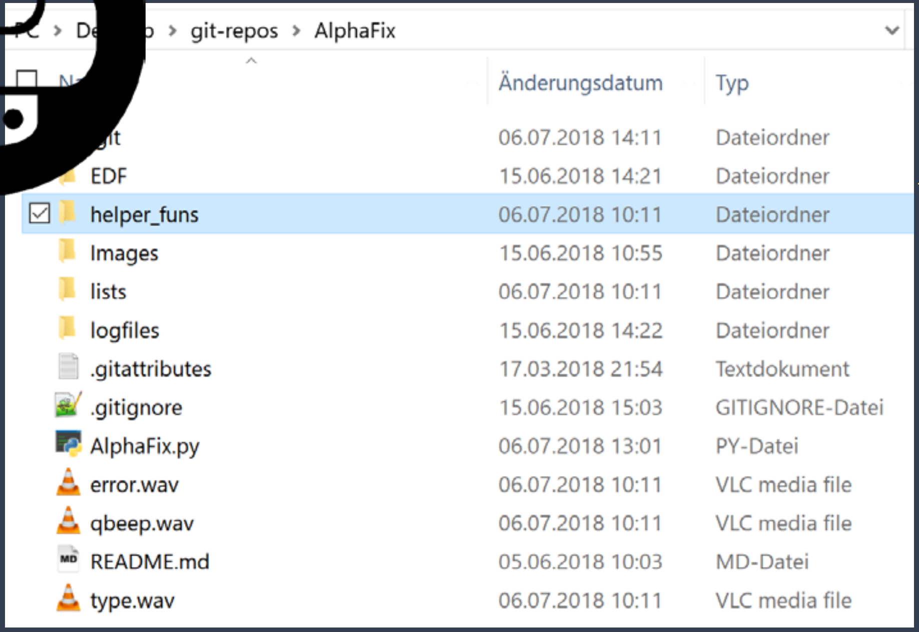
```
%Add Elektro-Pipe to path
addpath(genpath('C:\Users\wanja\Desktop\git-repos\unfoldilaus'));

%Add WM_utilities
addpath(genpath('C:\Users\wanja\Documents\MATLAB\WM_utilities'));

%Add wiedmann's erptools
addpath('C:\Users\wanja\Documents\MATLAB\erptools');
%add Memtoolbox
%addpath(genpath('C:\Users\wanja\Documents\MATLAB\MemToolbox'));
%add Unfold toolbox
addpath('C:\Users\wanja\Documents\MATLAB\unfold');
```

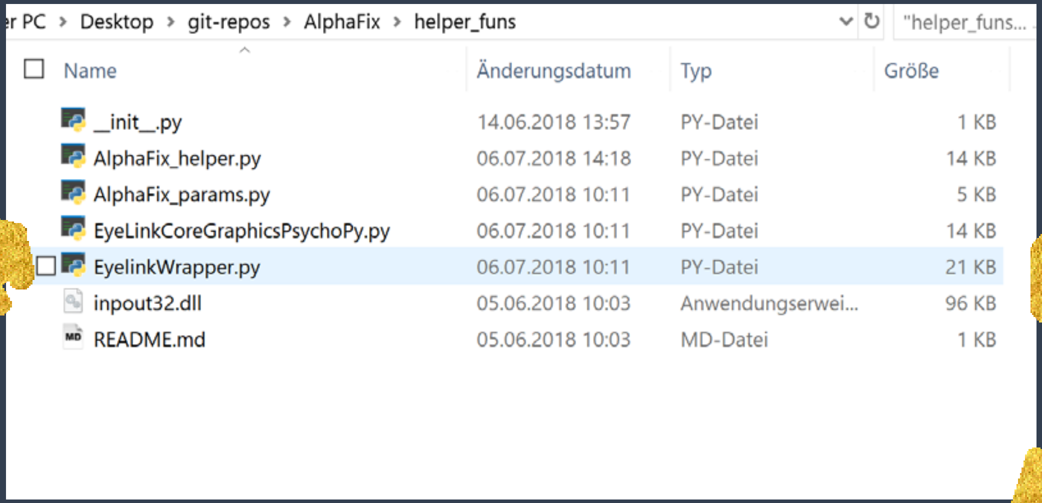
wm_ utilities

I‘ll only talk about Matlab – Same functions for Python!



C › De › git-repos › AlphaFix

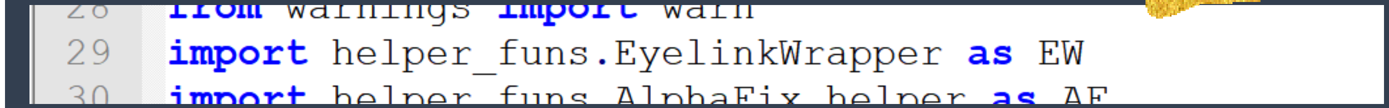
N	Änderungsdatum	Typ
git	06.07.2018 14:11	Dateiordner
EDF	15.06.2018 14:21	Dateiordner
☑ helper_funs	06.07.2018 10:11	Dateiordner
Images	15.06.2018 10:55	Dateiordner
lists	06.07.2018 10:11	Dateiordner
logfiles	15.06.2018 14:22	Dateiordner
.gitattributes	17.03.2018 21:54	Textdokument
.gitignore	15.06.2018 15:03	GITIGNORE-Datei
AlphaFix.py	06.07.2018 13:01	PY-Datei
error.wav	06.07.2018 10:11	VLC media file
qbeep.wav	06.07.2018 10:11	VLC media file
README.md	05.06.2018 10:03	MD-Datei
type.wav	06.07.2018 10:11	VLC media file



r PC › Desktop › git-repos › AlphaFix › helper_funs

"helper_funs...

□ Name	Änderungsdatum	Typ	Größe
__init__.py	14.06.2018 13:57	PY-Datei	1 KB
AlphaFix_helper.py	06.07.2018 14:18	PY-Datei	14 KB
AlphaFix_params.py	06.07.2018 10:11	PY-Datei	5 KB
EyeLinkCoreGraphicsPsychoPy.py	06.07.2018 10:11	PY-Datei	14 KB
□ EyelinkWrapper.py	06.07.2018 10:11	PY-Datei	21 KB
inpout32.dll	05.06.2018 10:03	Anwendungserwei...	96 KB
README.md	05.06.2018 10:03	MD-Datei	1 KB



```
from warnings import warn
import helper_funs.EyelinkWrapper as EW
import helper_funs.AlphaFix_helper as AF
```

Connect

```
[P, wPtr] = EyelinkStart(P, wPtr, “filename“)
→ doc EyelinkStart
```

P = your structure with parameters

can be empty, can contain extra config stuff

Always back assign P!

Does this:

- Starts and configures connection between EL and PC
- Opens a file to write to
- Defines sampling rate and what to store
- Defines locations of calibration points
- Nulls the TTL port
- Runs an initial calibration

Re-calibrate

```
EyelinkRecalibration(P)
```

Recalibrates the Eyetracker

Useful when:

- Break
- Subject moved a lot and eye is lost

Control Fixation

```
[didrecal, FixationOnset, hsmvd] =
EyelinkControlFixation(P, Tmin, Tmax, loc,
maxDegDeviation, eyelinkconnected, pixperdeg, dorecal,
Igoreblinks)
```

Highest possible level in wm_utilities

Does this:

- Takes a target location (loc)
- Subject has <Tmax> seconds to look at the target
- Subject has to look at target for <Tmin> seconds
- If not: start a recalibration (If you want that)
- Do you want to ignore blinks?

Get Gaze

```
[x, y, hsmvd] = EyelinkGetGaze(P, Ignoreblinks,
OverSamplingBehavior, targetXY,
FixLenDeg, eyelinkconnected, pixperdeg)
```

Lower level: Just get the gaze position and check whether that differs from the last position.

Does this:

- Checks if the Eyetracker has a new sample available
- If so, return the (x, y) pixel coordinate pair of the current gaze
- In addition, check if the current gaze differs more than <FixLenDeg> degree from targetXY. If so, <hsmvd> is true (1).

Behavioral data

```
[message] = EyelinkSendTabMsg(varargin)
```

stores behavioral data coregistered with time and ttl into the .edf file

Does this:

- Takes as many input arguments as you want
- Converts everything to strings
- Concatenates everything tab-delimited
- Prepends a “<“
- Sends <message> to .edf file

Stop recording

```
EyelinkStop(P, eyelinkconnected)
```

Finishes recording and puts Eyelink to standby

Does this:

- Stops the recording
- Copies the final file from the EL-host PC to the stimulation PC
- Then puts the Eyelink to standby

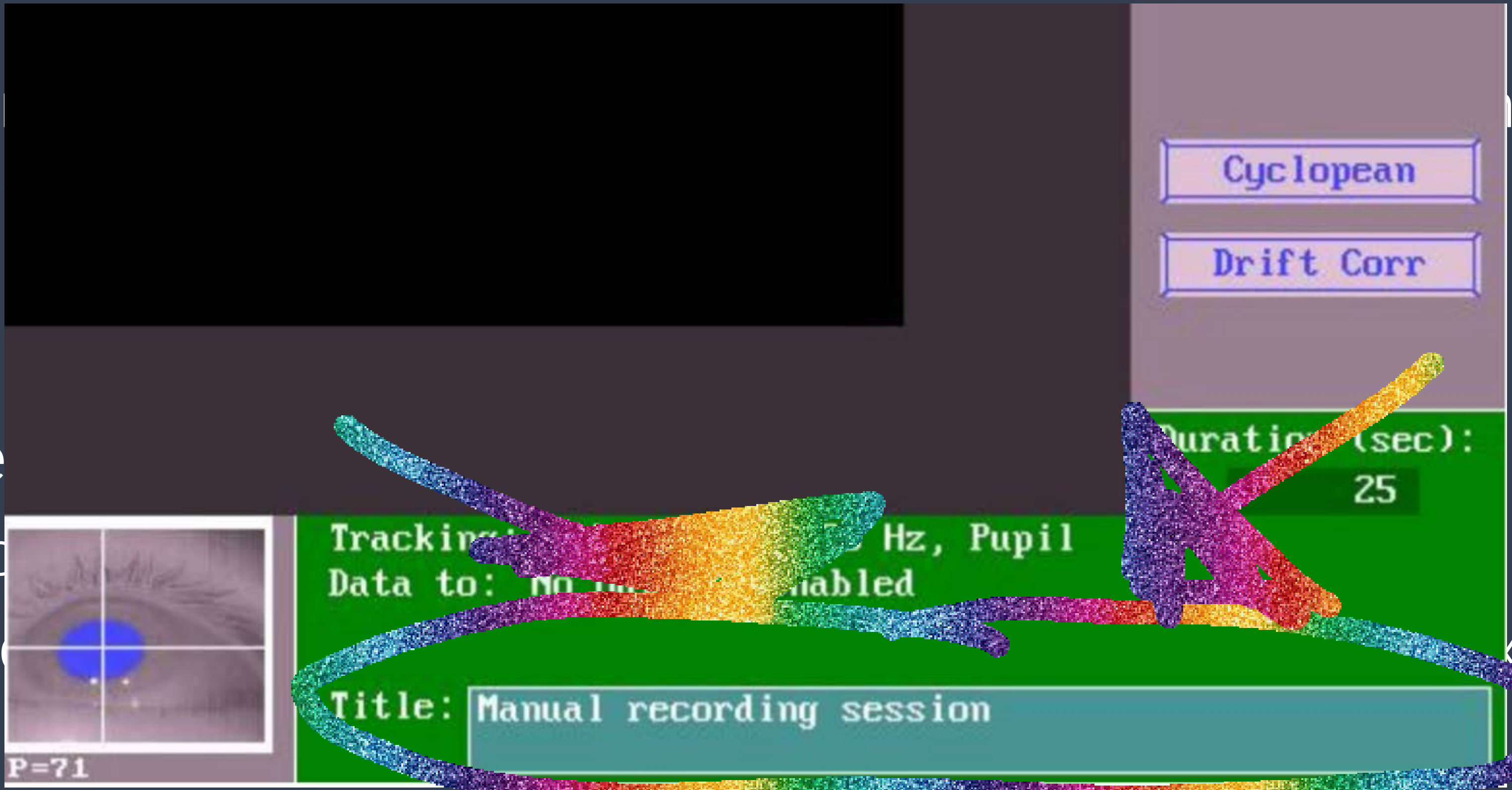
Status Notifier

`EyelinkNotify(message, eyelinkconnected)`

Lowe hat differs from the la

Does

- Che
- If so ze
- In a xLenDeg> deg



uratio (sec): 25

Trackin Hz, Pupil
Data to: No nabled

Title: Manual recording session

TTL trigger

`EyelinkAvoidWrongTriggers`

Prevents EEG-trigger-mess

Should be at the top of your non-eyetracking experiment

Does this:

- Checks if the Eyelink-host pc is turned on
- Throws an error if that‘s the case

Para- meters

‚eyelinkconnected‘ = boolean – if false, all functions are dummies

Dummymode doesn‘t always work for me, but it‘s worth trying

DIY

Implement Gazecontrol

DIY

- Open a terminal and run >`ptb3-matlab`
- Go to /home/tutorialbox/Documents/MATLAB/Tutorial/Task
- Start the experiment once by running >`TutorialExperiment.m`

DIY

- Add code to start and stop the Eyelink
- Display „Ruebezahl“ as status message, right after EyelinkStart
- Display the current trial number as status message
- Recalibrate after each break.
- Control that subjects look at the fixation symbol.
- Send „triggers“ (i.e., messages) after all relevant flips.
- Store the resulting behavioral data in the .edf

- If time: try to make the target‘s position gaze-dependent

Analyze

Preprocessing

What‘s possible ?

- Coregistration with EEG data:
  - Fixation Related Potentials → follow EYE-EEG/Elektro-Pipe
- Pupil Dilation
- Saccades
- Fixations
- Gazepaths
- …

→ Most stuff possible after following this template…

Parse Matlab

- `edf2asc`: binary available in the terminal (from the devpack)
- EYE-EEG
  - `parseyelink(file.asc)` ← converts .asc to .mat
  - `pop_importeyetracker(EEG, file.asc)` ← coregistration
- Elektro-Pipe
  - `func_importEye` combines all three and plots quality
  - Called by `prep01_preproc`

Parse R

- Jason Hubbard‘s `edfr` uses the devpack‘s C++ code directly
  - No need for `edf2asc`
  - Most sophisticated package for parsing (afaik)
  - `edf.all(), edf.recordings(), edf.events()`
  - = ET-data, Header, TTL

DIY

- Open Rstudio in the VirtualBox
- You‘ll find a file called `tutorial_preprocEDF.R`
- Let‘s first walk through that…

- Challenge:
  - Set the output time vector relative to fixation onset
  - Go to line 213. Try to add more info (as added in the experiment)
  - Add the new info to all necessary lines.