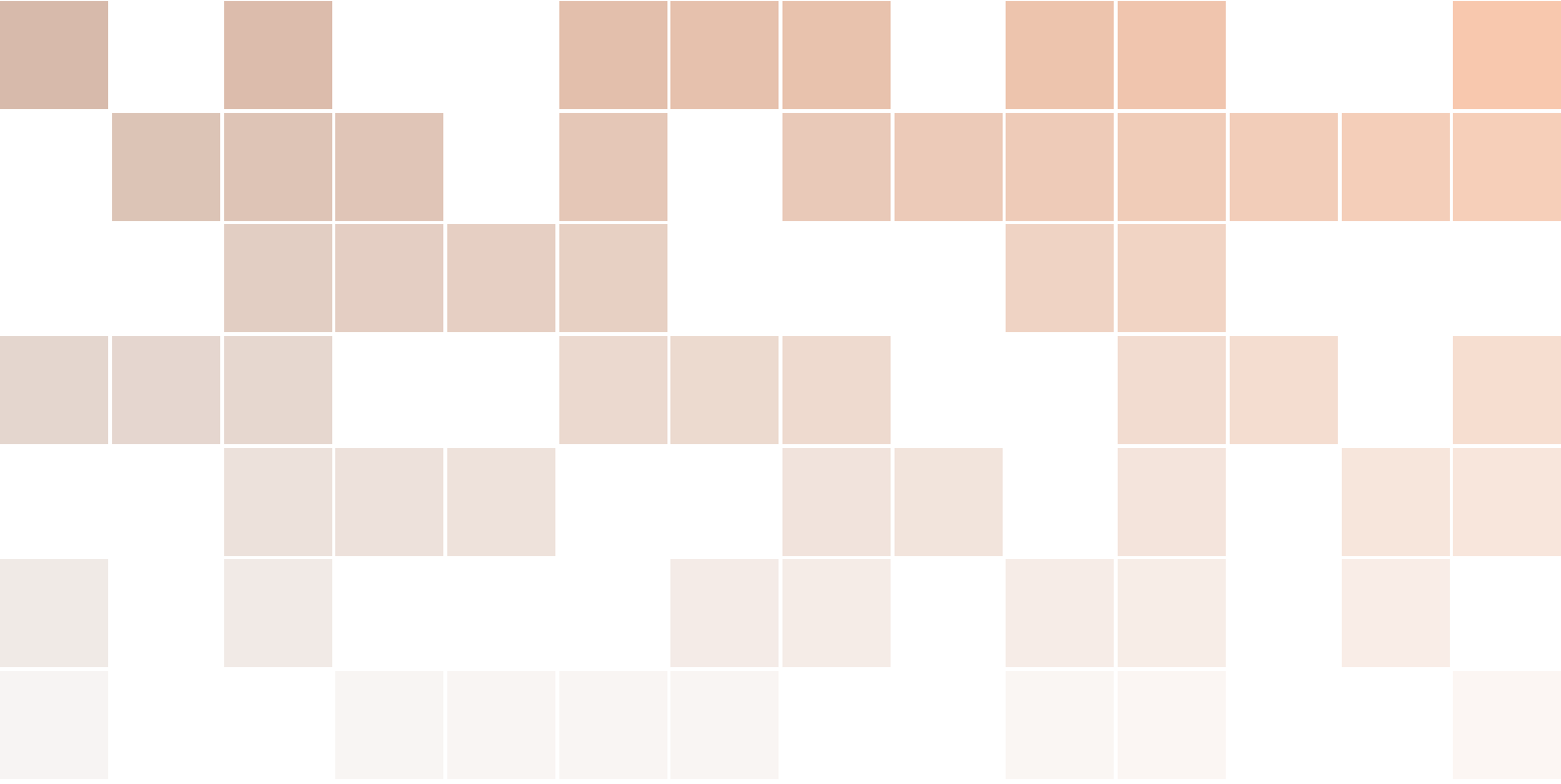




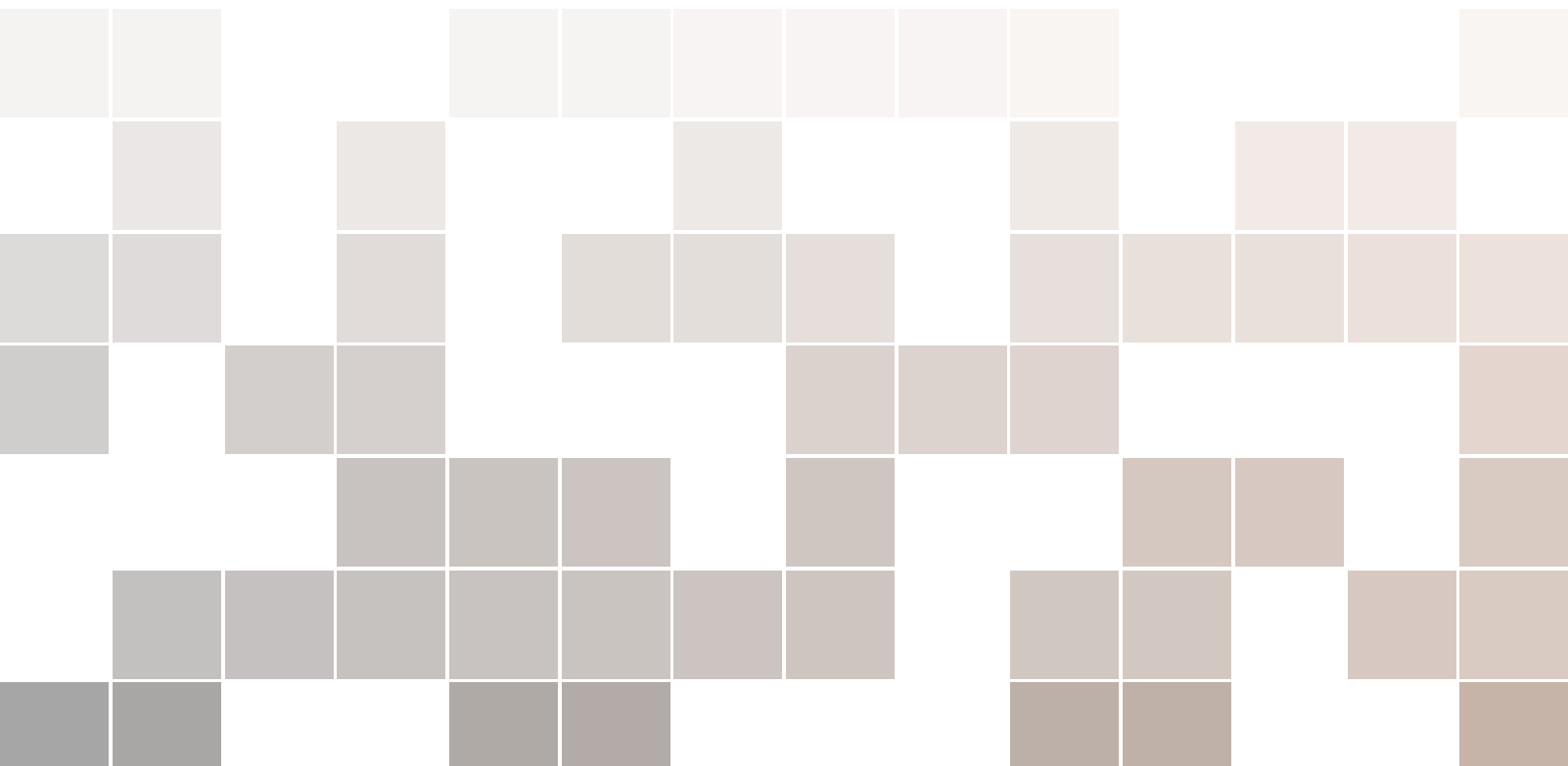
Pupil dilation toolkit for Matlab

Preprocessing and analysis

J.B.C.Marsman, PhD

NeuroImaging Center / Cognitive Neuro Psychiatry
Department of Neuroscience

University Medical Center Groningen



Copyright © 2014-2016 J.B.C. Marsman

UNPUBLISHED

[HTTP://MY-WEBSITE](http://MY-WEBSITE)

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

First printing, March 2016



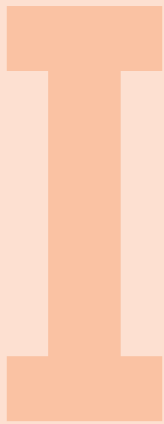
Contents

I	Background	
1	Introduction	7
1.1	Welcome	7
1.2	Intended audience	7
1.3	The human visual system	7
1.4	Viewing behaviour	9
1.5	Recording of eye movements	10
1.6	Pupillometry	11
II	Getting started	
2	Installation and Requirements	15
3	Overview	17
4	Illustrated examples	19
III	Preprocessing in depth	
IV	Analysis in depth	

5	Reference API	29
5.1	PDExperiment object	30
5.1.1	add_aggregate	30
5.1.2	askForBaseline(obj)	30
5.1.3	build_averages(e)	30
5.1.4	calculate_aggregate(e, index);	30
5.1.5	calculate_aggregates(e)	30
5.1.6	consolidateSettings(obj);	30
5.1.7	enable_selection_edit(e, type)	30
5.1.8	evaluate_selection(e, type)	30
5.1.9	export_data(e, file_out, varargin)	30
5.1.10	parse(settings,varlist);	30
5.1.11	export_stats(e, file_out, varargin)	30
5.1.12	parse(settings,varlist);	30
5.1.13	get_trials_based_on_label(e, session, trial_query)	30
5.1.14	getIndexFromSubjectName(e, list_in)	30
5.1.15	info(obj)	30
5.1.16	onClickPlotAxes(varargin)	30
5.1.17	onDataPointClick(e, unused, varargin)	30
5.1.18	PDExperiment(varargin)	30
5.1.19	plot(e)	30
5.1.20	plot_aggregate(e, index, highlighted)	30
5.1.21	plot_aggregates(e, table, event)	30
5.1.22	preprocess(obj);	30
5.1.23	remove_aggregate(e)	30
5.1.24	select_aggregate(varargin)	30
5.1.25	set_plot_sess(e)	30
5.1.26	set_plot_trial_edit(e)	30
5.1.27	set_plot_subject(e)	30
5.1.28	set_plot_trial_slider(e)	30
5.1.29	setPlotOptions(varargin)	30
5.1.30	setTrialStart(pattern)	30
5.1.31	togglePlotOptions(e)	30
5.1.32	update_aggregates(e);	30
5.2	PDSubject object	30
5.2.1	findAllfiles(obj, folder)	30
5.2.2	loadData(obj)	30
5.2.3	PDSubject(varargin)	30
5.2.4	preprocess(obj, settings)	30
5.2.5	rebuild(objs, varargin)	30
5.2.6	setFailSafePatternRelativeToPattern(objs, relativeTotype,type, timing)	30
5.2.7	setLabel(objs, type, pattern)	30
5.2.8	setLabelRelativeToLabel(objs, target, pattern, time)	30
5.2.9	setPattern(objs, type, pattern)	30
5.2.10	setPatternMulti(objs, type, pattern)	30

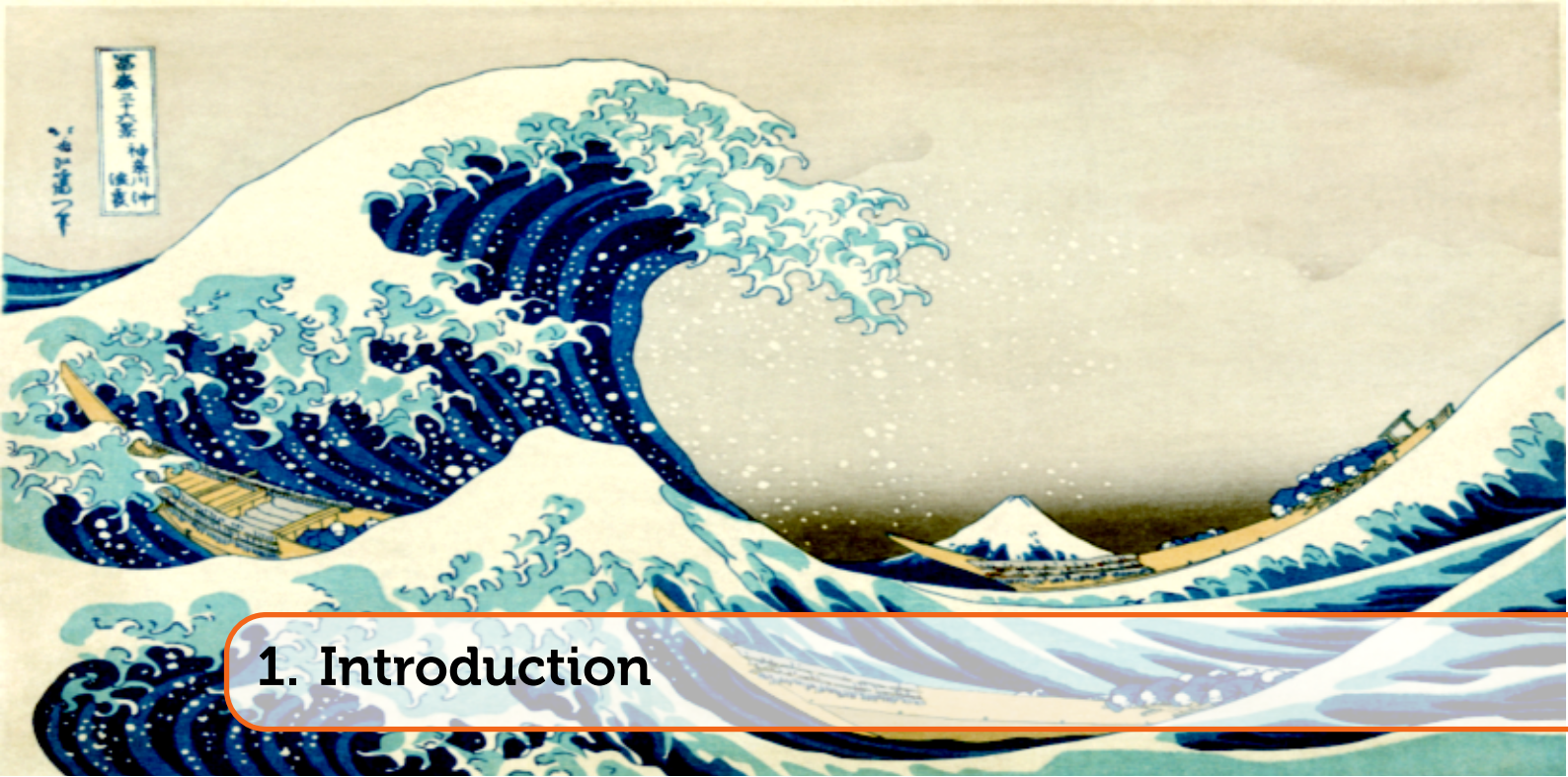
5.2.11	setPatternRelativeToPattern(objs, relativeTotype,type, timing)	30
5.2.12	setSettings(obj, settings)	30
5.2.13	summary(obj, level)	30
5.3	PDSession object	30
5.3.1	check_patterns(obj)	30
5.3.2	deconvolve(obj, settings)	30
5.3.3	display_and_log(obj, varargin)	30
5.3.4	importData(obj, varargin)	30
5.3.5	label(obj, params)	30
5.3.6	loadData(obj)	30
5.3.7	load_data_from_text(obj)	30
5.3.8	loadData(obj)	30
5.3.9	mean(session, varargin)	30
5.3.10	patternBuilder(session)	30
5.3.11	PDSession(varargin)	30
5.3.12	readEDF(filename)	30
5.3.13	plotTrialBasedonPattern(varargin)	30
5.3.14	preprocess(obj, settings)	30
5.3.15	readASC(obj)	30
5.3.16	rebuild(obj, params);	30
5.3.17	rebuildMulti(obj);	30
5.3.18	setFailSafePatternRelativeToPattern(obj, relativeTotype,type, timing)	30
5.3.19	setFramerate(obj)	30
5.3.20	setLabel(obj, pattern)	30
5.3.21	setLabelRelativeToLabel(objs, target, name, time)	30
5.3.22	setLabelRelativeToPattern(objs, target, name, time)	30
5.3.23	setPattern(obj, type, pattern)	30
5.3.24	setPatternMulti(obj, type, pattern)	30
5.3.25	setPatternRelativeToPattern(obj, relativeTotype, type, timing)	30
5.3.26	setSettings(obj, settings)	30
5.3.27	setTrialStart(obj, pattern);	30
5.4	PDTrial object	30
5.4.1	calculate_statistics(trials)	30
5.4.2	correct_for_baseline(obj, settings)	30
5.4.3	deconvolve(obj, settings)	30
5.4.4	remove_blinks(obj)	30
5.4.5	display_and_log(obj, varargin)	30
5.4.6	evaluate_model(obj, varargin)	30
5.4.7	getBaseline(trial, settings)	30
5.4.8	getTrialDataShiftedForMarker(trial, type, baseline_corrected, marker)	30
5.4.9	inpaint_nans.m	30
5.4.10	logtransform(obj)	30
5.4.11	PDTrial(varargin)	30
5.4.12	plot(varargin)	30
5.4.13	preprocess(obj, settings)	30
5.4.14	pupil_response_function(obj, t, n=10.1, t_max=930, f=1/(1024))	30
5.4.15	prf_convolve(obj, stick_model, slope)	30
5.4.16	pupil_response_function(obj, t, varargin)	30
5.4.17	remove_blinks(obj, settings)	30
5.4.18	setSettings(obj, settings)	30
5.4.19	stick_model(obj, varargin)	30

Bibliography	31
Books	31
Articles	31
Index	33



Background

1	Introduction	7
1.1	Welcome	
1.2	Intended audience	
1.3	The human visual system	
1.4	Viewing behaviour	
1.5	Recording of eye movements	
1.6	Pupillometry	



1. Introduction

1.1 Welcome

Thank you for taking the time and effort to read this manual for the PD analysis toolkit. At our institute a number of pupil dilation studies were conducted and no existing code was available at the time. Although some code earlier analyses borrowed software from the Psychology department, the support I had to provide on this software – written in R – did not feel comfortable. Therefore, I decided to transfer the software to Matlab and extend it for further analysis.

1.2 Intended audience

The toolkit and software is written for researchers interested in pupil dilation research. For many researchers, there is a relatively high threshold in analysis due to lack of programming skills. This PD analysis toolkit should provide a simplistic interface to complex analysis of pupil dilation. The following sections provide an overview of visual research of eye movements.

1.3 The human visual system

Our visual system comprises a number of complex structures, each having their specific function. Figure ?? displays all major structures of the visual system. The incoming light first passes the pupil and is projected on the retina (left bottom part of figure ??). Light sensitive cells in the retina, called receptors, are specialized for light perception at either low luminance (rods) or at high luminance (cones) and convert the light signal into a neural signal. The human retina contains three different types of cones that each have their maximum sensitivity in a different part of the visible spectrum. Because of this, the cone signals provide the basis for human color vision.

The retina contains a number of additional layers of cells, which also perform the first processing on the neural signal. Specialized ganglion cells exist that relay specific aspects of the visual information to the brain. The main classes of retinal ganglion cells are Magnocellular (M), Parvocellular (P) and Koniocellular (K) cells. M-cells combine the information from the

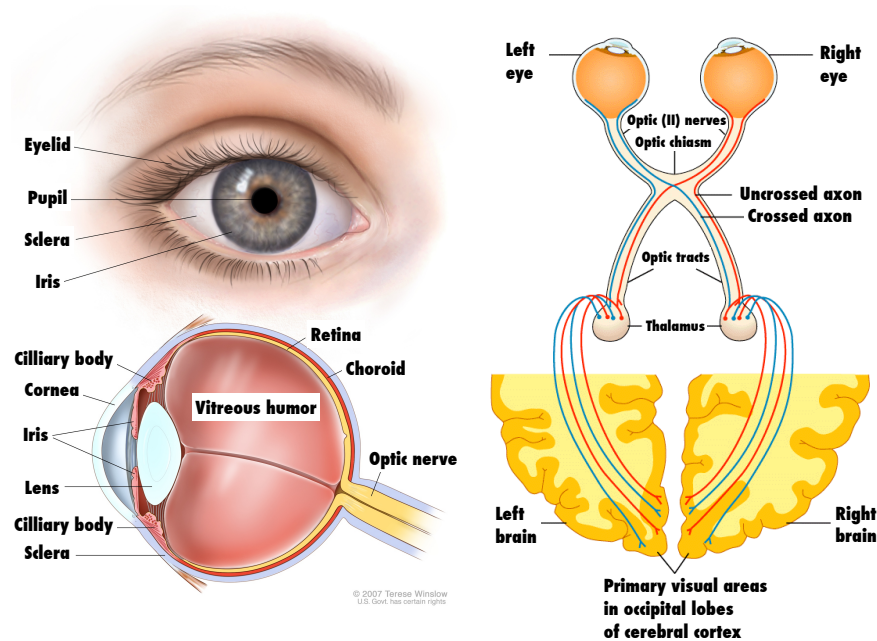


Figure 1.1: The visual system.

three different cones, and mainly processes luminance and contrast information. M-cell signals are used in the processing of movement and are highly sensitive to differences in brightness. P-cells process color signals from the cones in the retina. To date, the precise function of the K-cells remains unclear, but they may also be involved in color processing. From the retina, the neural signals are relayed via the optic nerve, optic chiasm, Lateral Geniculate Nucleus (LGN), and the optic radiations to the primary visual cortex (V1) in the occipital pole of the cerebral cortex (see figure ??, right). V1 deals with low level processing of visual input (e.g., orientation or color). Furthermore, V1 has a retinotopic organization. Retinotopy entails the projection of the visual input where adjacent points in the visual field correspond to adjacent points on the cortex. From V1 onward, neural signals are relayed to many different cortical and subcortical structures. The subsequent processing of visual information is organized into two major processing pathways.

These two pathways relay information from visual cortex towards the rest of the brain. After their discovery in 1967 by Trevarthen in frogs and in fish by Ingle, a large body of literature has corroborated the existence of these pathways in both mammals and humans [Trevarthen68, Mishkin1982, Milner1996, Milner2008]. Each pathway serves a different function in visual information processing.

The dorsal pathway (the "where" pathway) enables perception for action, and runs from the occipital pole towards the parietal lobe. The dorsal route primarily processes information from the magnocellular ganglion cells. Information in the dorsal route has been shown to contribute to visuospatial functions (e.g., grasping, shifting attention or reaching). Priority reports that regions in the dorsal stream contribute to the processing of visuospatial attention in natural human viewing behaviour.

The ventral pathway (the "what" pathway) enables perception for recognition, and runs from the occipital pole towards temporal lobes. The ventral pathway is thought to process primarily input from the parvocellular cells. This pathway deals with visual processing for conscious perception of objects and the ability to recognize and identify them. In contrast, ventral regions have also been identified to rapidly extract the gist of a scene. Chapter 6

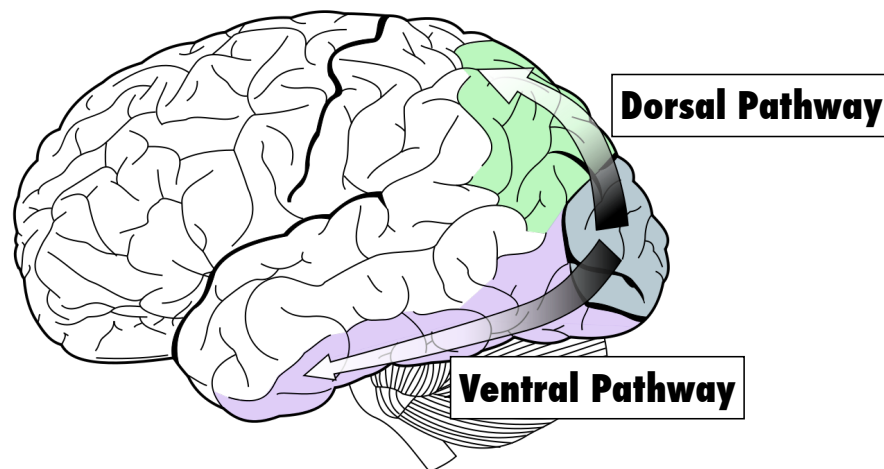


Figure 1.2: Two visual pathways each enabling different visual functions: Perception for action in the dorsal stream (green); Perception for recognition in the ventral stream (purple). Source: <http://www.wikipedia.org>

examines how both types of visual perception are combined in the ventral visual system for natural human viewing behaviour.

1.4 Viewing behaviour

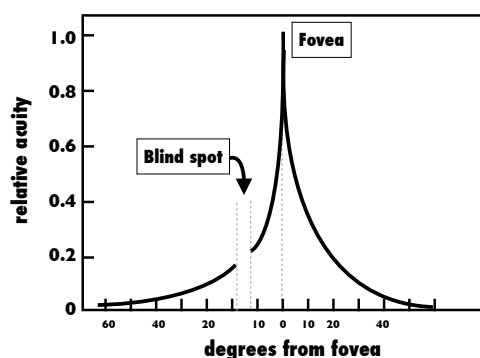


Figure 1.3: Distribution of relative visual acuity across the retina. Recreated after Core1978 [Core1978]

Human vision is only crisp at the center of gaze and visual acuity rapidly decreases in the periphery of the visual field. Therefore, we need to actively move our eyes to change our focus when we want to scrutinize different scene elements. During everyday behaviour, we move our eyes 3-5 times per second [Henderson2003]. Eye movements can be categorized in a number of different behaviours:

Fixations During a fixation, gaze is maintained on a single location. In general a fixation can last between approximately 100 - 2000 ms and on average lasts 300 ms.

Saccades Rapid movements of the eye to redirect our gaze.

Smooth pursuit eye movements During a smooth pursuit eye movement, the eye closely follows a moving object.

The essential role of eye movements in visual perception has been repeatedly demonstrated [Findlay1998, Hayhoe2005, MartinezConde2004]. When the retinal image is stabilized, vision quickly degrades due to adaptation -the decreased responsiveness to a constantly present stimulus. Eye-movements are necessary to restore vision [MartinezConde2004]. But also many tasks require that we continuously shift our gaze to acquire information from different aspects of the environment. The change of gaze results in novel samples of the environment, each time projected differently on the retina. This sampling of information from the environment requires that visual perception is investigated within the framework of active vision [Hayhoe2005, Henderson2003, Findlay1998, Einhauser2010]. The work in this thesis contributes to the investigation of active vision using fMRI.

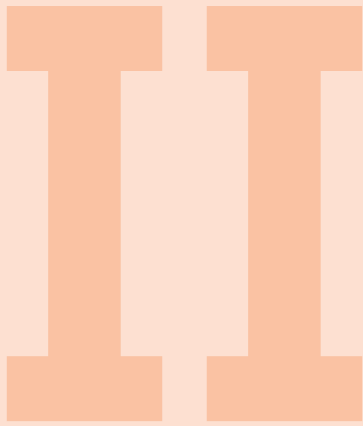
1.5 Recording of eye movements

Most of the experiments described in this thesis use eye tracking as one of its core methods to investigate human visual behaviour. Mainly three types of eye trackers exist: 1) sensor-based eye trackers, where a magnetic sensor embedded to a contact lens is placed on the eye. 2) Electro-oculography (EOG), which uses electrodes placed around the eyes to register eye position. 3) Video-based eye tracking, where the image of the eye is captured using a camera. Because video-based eye tracking is non-invasive, it is often preferred and most widely used in research as well as commercial applications.

In its essence, video based eye trackers comprise a camera unit with illumination. Today's eye tracking systems use an infrared (IR) illumination produced via light-emitting diodes (LEDs). There are two reasons for this type of lighting in eye tracking systems: First, the use of a light source in the scene enhances the quality of the image in terms of contrast and intensity levels, and it facilitates detailed image analysis and gaze estimation. Second, IR optical radiation is invisible to the human eye, so that the lighting is comfortable and does not distract user's attention. The camera unit retains both high temporal and spatial resolution. A close-up video image of the eye is sent to a computer, on which a dedicated image processing algorithm extracts the pupil and the bright reflection of the cornea to determine its position and subsequently stores the positions to a file. Prior calibration of the eye tracking system is necessary to map the eye position onto screen coordinates. This is achieved by recording the position of the pupil together with the position of the reflection of the cornea for a number of fixed points on the screen.

After an experiment has ended, the researcher is left with a list of screen coordinates and their corresponding time stamps. However, eye movements events of human viewing behaviour are fixations, saccades and blinks. Event detection is used to classify recorded gaze points into periods of fixation, saccade, smooth pursuit, blink, and noise. These algorithms primarily focus on either fixations or saccades [Salvucci2000]. Fixation Detection algorithms assess whether dwell time on certain locations with minimal dispersion exceeds a certain threshold. If this is the case, a fixation is detected. Saccade detection algorithms are driven by the increase in the eye's velocity to determine saccades.

1.6 Pupillometry



Getting started

2	Installation and Requirements	15
3	Overview	17
4	Illustrated examples	19



2. Installation and Requirements

3. Overview

There are a number of basic building blocks inside the pupil dilation toolkit (see Figure 3.1). These are:

PDExperiment object holds all subject inside the experiment, and allows for (equal) pre-processing and analysis of all subjects in the array.

PDSubject object may contain multiple sessions, stores information regarding the subject.

PDSession object contains all trials and provides information regarding the asc file which is imported.

PDTrial object is the lowest object in hierarchy. PDTrial stores the actual traces (both raw and preprocessed), markers, labels and other trial related information.

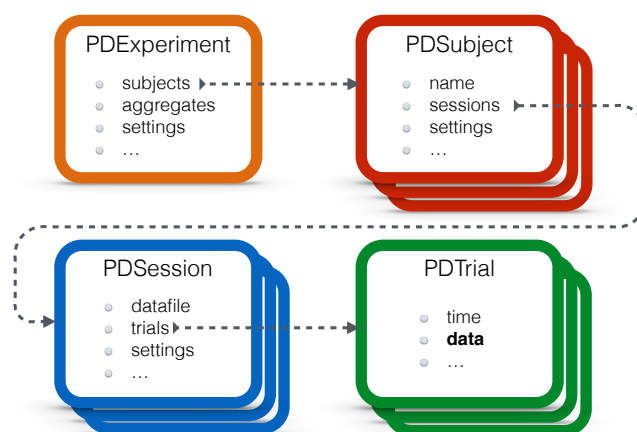


Figure 3.1: Objects and their relations to one another as used inside PD toolkit

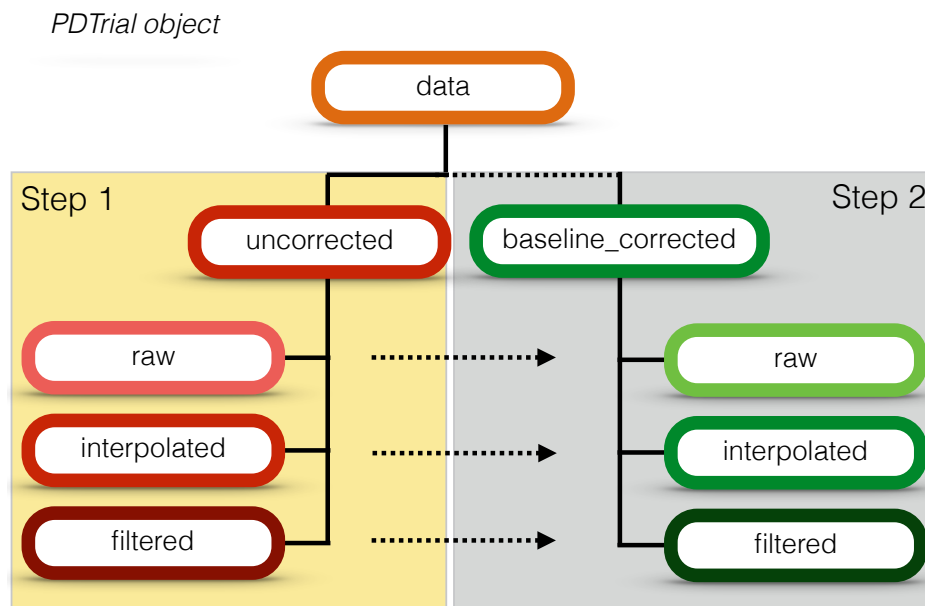


Figure 3.2: Inside *PDTrial* object, data is stored in many stages.

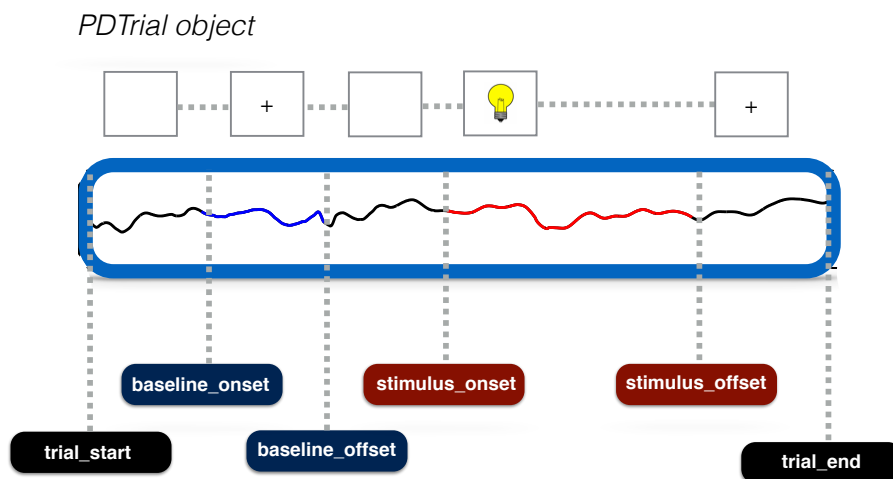


Figure 3.3: Example trial with markers specified



4. Illustrated examples

This simple example shows the use of PD objects:

Code snippet 4.1

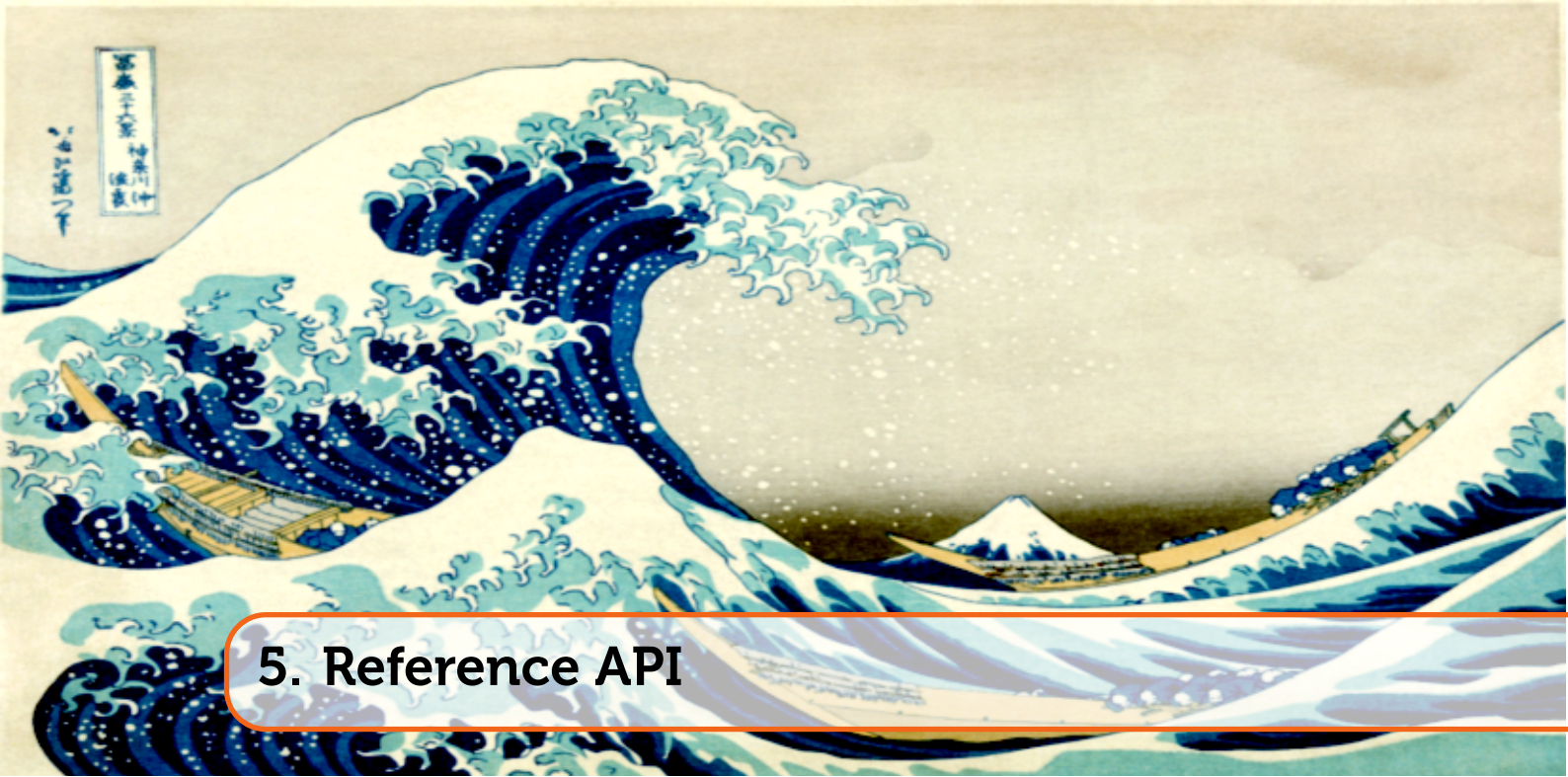
```
s = PDSubject('C:\\myExperiment\\Subject1\\');
```





Reference functions

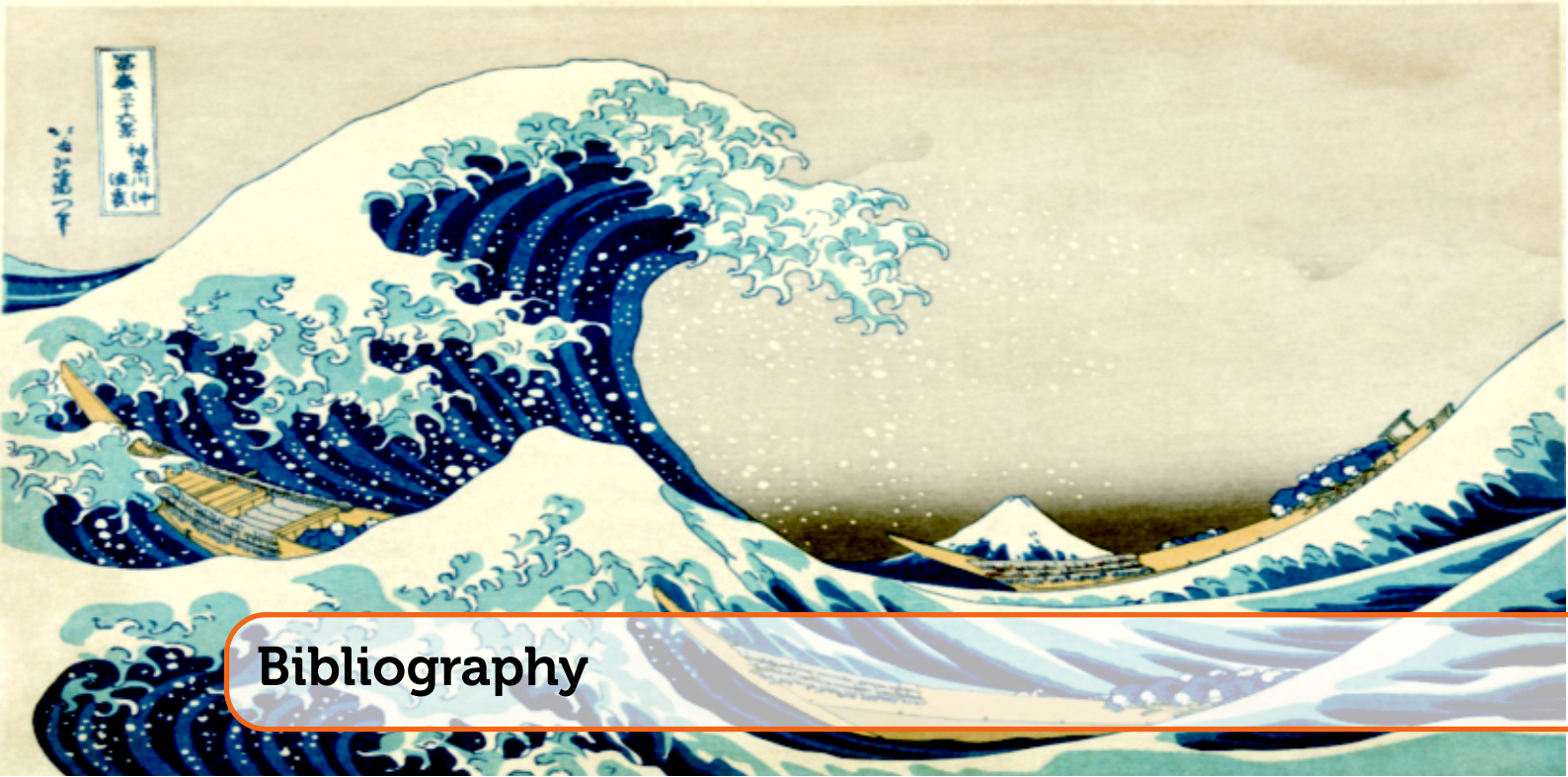
5	Reference API	29
5.1	PDExperiment object	
5.2	PDSubject object	
5.3	PDSession object	
5.4	PDTrial object	
	Bibliography	31
	Books	
	Articles	
	Index	33



5. Reference API

5.1 PDExperiment object

- 5.1.1 add_aggregate
- 5.1.2 askForBaseline(obj)
- 5.1.3 build_averages(e)
- 5.1.4 calculate_aggregate(e, index);
- 5.1.5 calculate_aggregates(e)
- 5.1.6 consolidateSettings(obj);
- 5.1.7 enable_selection_edit(e, type)
- 5.1.8 evaluate_selection(e, type)
- 5.1.9 export_data(e, file_out, varargin)
- 5.1.10 parse(settings,varlist);
- 5.1.11 export_stats(e, file_out, varargin)
- 5.1.12 parse(settings,varlist);
- 5.1.13 get_trials_based_on_label(e, session, trial_query)
- 5.1.14 getIndexFromSubjectName(e, list_in)
- 5.1.15 info(obj)
- 5.1.16 onClickPlotAxes(varargin)
- 5.1.17 onDataPointClick(e, unused, varargin)
- 5.1.18 PDExperiment(varargin)
- 5.1.19 plot(e)
- 5.1.20 plot_aggregate(e, index, highlighted)
- 5.1.21 plot_aggregates(e, table, event)
- 5.1.22 preprocess(obj);
- 5.1.23 remove_aggregate(e)
- 5.1.24 select_aggregate(varargin)
- 5.1.25 set_plot_sess(e)
- 5.1.26 set_plot_trial_edit(e)
- 5.1.27 set_plot_subject(e)
- 5.1.28 set_plot_trial_slider(e)
- 5.1.29 setPlotOptions(varargin)



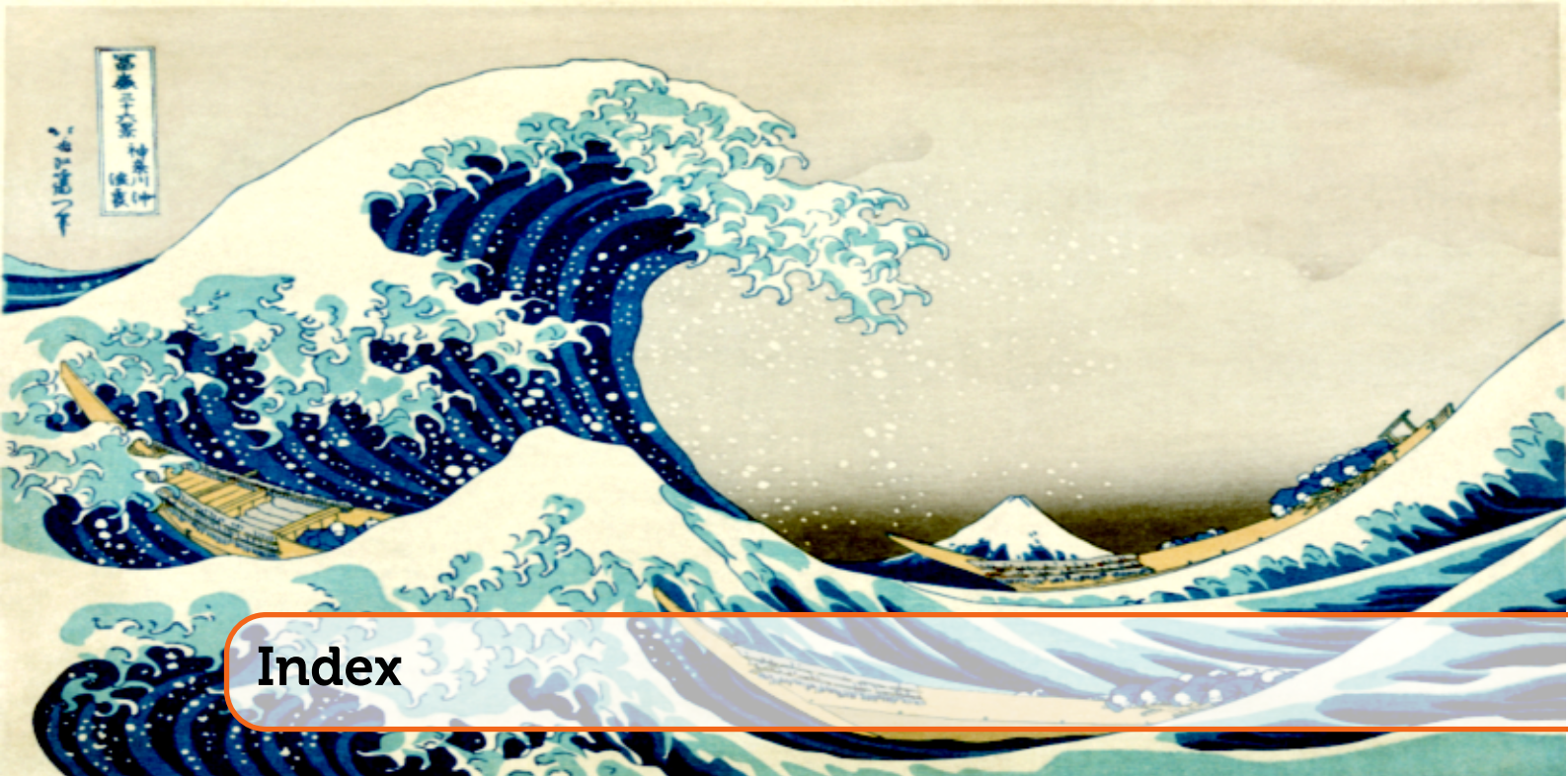
Bibliography

Books

[Smi12] John Smith. *Book title*. 1st edition. Volume 3. 2. City: Publisher, Jan. 2012, pages 123–200.

Articles

[Smi13] James Smith. “Article title”. In: 14.6 (Mar. 2013), pages 1–8.



Index

C

Citation.....8
Corollaries.....10

D

Definitions.....9

E

Examples.....10
 Equation and Text.....10
 Paragraph of Text.....11
Exercises.....11

F

Figure.....15

L

Lists.....8
 Bullet Points.....8
 Descriptions and Definitions.....8
 Numbered List.....8

N

Notations.....10

P

Paragraphs of Text.....7
Problems.....11
Propositions.....10
 Several Equations.....10
 Single Line.....10

R

Remarks.....10

T

Table.....15
Theorems.....9
 Several Equations.....9
 Single Line.....9

V

Vocabulary.....11