Eye Tracking

Applications, Recording, Analytics, Interaction

Andrew T. Duchowski  duchowski@clemson.edu
Clemson University
School of Computing
100 McAdams Hall
Clemson, SC, 29634
USA

© 2023 Andrew T. Duchowski
Acknowledgements

Funding
U.S. National Science Foundation

Donations

Collaborations
SWPS, Universität Tübingen
Collaborators

SWPS University of Social Sciences and Humanities
Krzysztof & Iza Krejtz
Justyna Garnier

Universität Tübingen
Nina Gehrer
Objectives

The tutorial is based on four objectives: a survey of the field followed by hands-on examples of gaze recording, analytics and gaze interaction. The goals are for attendees to learn about classic eye-tracking research as well as current directions and open problems, and then to learn about basic and advanced eye movement analysis and interaction techniques.

For 2023, the tutorial will include new PsychoPy demonstration code with which attendees will be able to record their own data provided they have access to an eye tracker supported by PsychoPy, i.e., Tobii, Gazepoint, SR Research, or simulate gaze data with the mouse (also an available PsychoPy option).
Abstract

The tutorial starts with an overview of eye-tracking applications, distinguishing eye movement analysis from synthesis in virtual reality, games, and other venues including mobile eye tracking. The focus is on five forms of applications: diagnostic (off-line measurement), active (selection, look to shoot), passive (foveated rendering, a.k.a. gaze-contingent displays), assistive (translation), and expressive (gaze synthesis). The tutorial then covers details of a Python-based gaze analytics pipeline developed and used by Prof. Duchowski and others. The gaze analytics pipeline consists of Python scripts for extraction of raw eye movement data, analysis and event detection via velocity-based filtering, collation of events for statistical evaluation, analysis and visualization of results using R. The tutorial covers basic eye movement analytics, e.g., fixation count and dwell time within AOIs, as well as advanced analysis using ambient/focal attention modeling and gaze transition entropy. Newer analytical tools and techniques such as microsaccade detection and the pupillary activity will be covered, time permitting. The tutorial concludes with an overview and demo of how to build an interactive Python application.
Taget Audience

The intended audience should largely be composed of novice attendees who may have heard something about eye tracking and would like to learn more about what is involved.

Seasoned veterans, are also welcome, they may benefit from the section on experimental design.

Finally, time permitting, anyone interested in developing a basic eye-tracking application will benefit from an overview of a Python application designed to obtain gaze information in real-time from a commodity eye tracker.
• 80 min: Gaze Applications
  - 20 min: Active
  - 20 min: Passive
  - 20 min: Expressive
  - 20 min: Diagnostic
• 100 min: Gaze Analytics
  - 50 min: Experimental Design & Analytics Pipeline
  - 50 min: Gaze Metrics & Statistics
• 50 min: Gaze Recording
• 30 min: Gaze Interaction
  - 10 min: Instrumentation
  - 10 min: Socket Communication
  - 10 min: Real-time Gaze Signal Analysis
Dr. Duchowski is a professor of Computer Science at Clemson University. He received his baccalaureate (1990) from Simon Fraser University, Canada, and doctorate (1997) from Texas A&M University, USA, both in Computer Science. His research and teaching interests include visual attention and perception, computer vision, and computer graphics. He is a noted research leader in the field of eye tracking, having produced a corpus of related papers and a monograph on eye tracking methodology, and has delivered courses and seminars on the subject at international conferences. He maintains Clemson's eye tracking laboratory and teaches a regular course on eye tracking methodology attracting students from a variety of disciplines across campus.
Biosketch

• Professor of Computer Science
  - *Gastprofessor* 2016 ETH Zürich
  - PhD 1997 Texas A&M
  - BSc 1990 Simon Fraser

• Visual Computing division at Clemson
  - computer graphics, computer vision, eye tracking

• Founded ETRA, CHI alt.chi, and SIGGRAPH

*Groovy Graphics Assignments*
Eye Tracking
Research & Applications

Andrew T. Duchowski  duchowski@clemson.edu
Eye Tracking

Research & Applications

assistive

active

selection/look to shoot

translation/interpretation

passive

foveated rendering (gaze-contingent displays)

expressive

gaze synthesis

diagnostic (off-line)

assessment/training
Intro: binocular eye tracking in VR

- Recent entrants: ~$600
  - Facebook (bought Eye Tribe, owns Oculus)
  - Google (bought Eyefluence)
  - Apple (bought SMI)
  - FOVE
  - Samsung or HTC Vive (using SMI)
  - HTC Vive (using Pupil Labs)
Intro: binocular eye tracking in VR

- Earlier technology: ~$60,000
  - VR Research HMD
  - ISCAN eye tracker
  - 2 IR LEDs per eye
  - half-silvered mirrors

Duchowski, et al. (ETRA 2002)
Who is left today?

- **Wearables**
  - Ergoneers (Dikablis), Pupil Labs, Tübingen (EyeRecToo), Tobii (Glasses2)

- **Desktop**
  - SR Research (Eyelink), Gazepoint (GP3), Tobii (various)
Affordable to facilitate classroom instruction

• Clemson’s eye tracking classroom houses 20 Gazepoint GP3 trackers
Emergent Technology

- AR
  - Microsoft’s HoloLens 2

- Pupil Labs Invisible
Emergent Technology

• AR
  - Meta Oculus Pro
Reliability of Eye Trackers

- Web-based: limited accuracy but ok (see our ETRA 2022 paper)
- Video-based: accurate, sampling rate determines event detection
  - 300 Hz needed for microsaccades
- New technologies: MEMS, photosensors
- Fixational drift:
  - something else entirely, more gaze dynamics than anything
  - current fixational algorithms would ignore this (wrongly)
The basics

Schematic representation of the human visual field.

From Sundstedt, V., Gazing at Games, SIGGRAPH 2010 course notes. Adapted from MacEvoy, B. The Visual Field. http://www.handprint.com
Saccades

- Used to shift gaze
  - < 1 deg to > 45 deg
  - 500 deg/s
Smooth pursuits

- Used to track object in motion
Vestibular Ocular Response

- Used to counter-rotate eyes when head turns
Vergence

- Used for depth perception (arm’s length)
Fixations

- Used to stabilize gaze
  - characterized by tremor, drift, and microsaccades
  - 90% viewing time spent in fixations

from Pritchard (1961)
Fixations

- Silicon sees a cat
  - time-derivative silicon retina (see youtube vid)
  - image fades when no movement

Mahowald and Mead (Scientific American, 1991)
Eye Tracking

Research & Applications

- assistive
  - selection/look to shoot
  - translation/interpretation
- active
  - foveated rendering (gaze-contingent displays)
- passive
  - gaze synthesis
- expressive
  - assessment/training
- diagnostic (off-line)
Classic gaze-based selection

- Key problem: Midas Touch
  - solution: dwell time

Jacob (CHI 1990)
Gaze-based PIN selection

Best and Duchowski (ETRA 2016)

dwell-time

boundary-crossing gestures
Gaze-based selection in gameplay

• Gaze-based viewpoint orientation

Smith and Graham (ACE 2006)
Gaze-based selection in gameplay

- Eye-based pointing

Smith and Graham (ACE 2006)
Tobii integrated eye tracker

* Note: I don’t get paid by Tobii
Summary: Active gaze

Gaze interaction in the post-WIMP world:
SIGCHI 2013 workshop

Bob Jacob, Raimund Dachselt, Andreas Bulling, me, Sophie Stellmach, Veronica Sundstedt

- **Outstanding problems:**
  - Midas Touch ever present
  - multi-modality gaining popularity
  - gaze gestures also popular
  - new interaction techniques always welcome
Summary: Active gaze

- **Challenges:**
  - privacy
  - gaze + X
  - recommendation & guidance
  - intent & prediction

- **Scenarios:**
  - mobility
  - healthcare
  - play & learn
  - everyday use

Ubiquitous Gaze Sensing and Interaction: Dagstuhl 2018 seminar 18252

L. L. Chuang, A. T. Duchowski, P. Qvarfordt, D. Weiskopf
Eye Tracking

Research & Applications

assistive
active
passive
expressive
diagnostic (off-line)

selection/look to shoot
foveated rendering
(gaze-contingent displays)
gaze synthesis
assessment/training

translation/interpretation
Foveated rendering for gaze-tracked VR

- Michael Abrash of Oculus Research:
  - foveated rendering core VR technology in 5 years
  - “great eye tracking is central to the future of VR”
Foveated rendering for gaze-tracked VR

- Perceptually-based foveated VR from NVIDIA

Patney et al. (SIGGRAPH Asia 2016)
Foveated 3D graphics

• Rendering speedup
  - 100 times speedup predicted
  - at 70 deg field of view
  - three layers:
    - foveal
    - middle
    - outer

Guenter et al. (SIGGRAPH Asia 2012)
Simulation of arbitrary visual fields

• Gaze-contingent rendering can simulate glaucoma
• Arbitrary visual field in general simulates scotoma
Simulation of arbitrary visual fields

- Simulating scotoma
- Spatiochromatic degradation

Duchowski and Eaddy (EG short papers, 2009)
Duchowski and Çöltekin (TOMCCA, 2007)
Duchowski et al. (TAP, 2009)
Simulation of arbitrary visual fields

- Pixel-based approach using Mip-mapping
- GLSL shaders make this fairly easy

Duchowski (ETRA 2004)
Saccade prediction

Arabadzhyska et al. (SIGGRAPH 2017)
Saccade prediction

Arabadzhiyska et al. (SIGGRAPH 2017)
Stereo zone of comfort

- Accommodation-vergence conflict

Shibata et al. (2011)
Stereo zone of comfort

- Desktop and Wheatstone haploscope stereo displays
Seamless disparity remapping

- Remapping disparity at gaze location
  - e.g., if disparity range is narrow

Kellnhofer et al. (SIGGRAPH 2016)
Gaze-contingent DOF

- Compute DOF following Mantiuk et al. (2011)
  - Circle of Confusion (CoC) determines peripheral blur
  - implement on GPU as per Riguer et al. (2004)
  - substitute depth value ($d_0$) by gaze depth $z$

\[
\text{CoC} = a \cdot \left| \frac{f}{d_0 - f} \right| \cdot \left| 1 - \frac{d_0}{d_p} \right|
\]
Computing gaze depth

\[ \Delta x = x_r - x_l \]

\[ z = \frac{\Delta x D}{\Delta x - a} \]

Daugherty et al. (ETRA 2010)
Computing gaze depth

Duchowski et al. (ETRA 2014)
Computing gaze depth

Duchowski et al. (ETRA 2014)
Render for comfort, not speed
Gaze-contingent DOF

Duchowski et al. (SAP 2014)
Focal surface displays

Matsuda et al. (SIGGRAPH 2017)

Oculus Research
Focal surface displays
Summary: Passive gaze

- Outstanding problems:
  - eye is not rigid (like a bag of jello)
  - need lots of IR illuminators (in VR, according to Oculus)
  - latency, latency, latency (e.g., saccade prediction)
For more information...

- Various sources, incl. proceedings
  - SIGGRAPH, ETRA, ETVIS, etc.
For more information...

- Michael Abrash’s blog is also a good read

Eye motion in VR
Motivation
Motivation

Jeff Wadlow’s *Kick-Ass 2* (2013) © Universal
Uncanny valley

familiarity

human likeness

50%

100%

moving

still
Motion in games

- Team Bondi’s *L.A. Noire* Non-Player Characters:
  - eye motions are key to player deciding if NPC is lying
Gaze and trust

- Gaze affects perceived trust (Normoyle et al., 2013)
Bidirectional gaze simulation

- Stochastic model promotes gaze coordination (Andrist, 2017)
Avatar realism

- Garau et al. (2003)
  - found strong interaction between character realism and gaze
  - a more realistic character requires more elaborate gaze
Self-avatar realism

- Borland et al. (2013)
  - avatar’s rotating eyes promoted eye contact vs. fixed gaze
  - increases self-identification with the avatar
All you need is an eye tracker

* Note: I still don’t get paid by Tobii
All you need is an eye tracker

• Not always easy to use, noisy
• They don’t always provide reference video
  - hence signal attenuation vs. reference not always possible
Motion capture

• Automatic generation of realistic eye motion
• Why not just use real data?
  - humans or eye trackers not always available

cleaned data  eye tracker camera
Synthesizing eye movements

- Lee et al.’s (2002) *Eyes Alive*
  - top-down approach modeling the saccadic *main sequence*
    \[ \Delta t = 2.2\theta + 21 \]

- We approach simulation bottom-up
  - start with a signal processing perspective
Coupling eye & head movements

- Yeo et al.’s (2012) Eyecatch
  - saccades, smooth pursuits
  - head coupling
  - no microsaccades
Emotional expressivity

- Li and Mao (2012):
  - saccades, blink rate, pupil diameter
  - matching Facial Action Coding System and Geneva Emotion Wheel
Probabilistic eye movements

- Lance and Marsella (2008)
  - combined eye movements with head rotations
  - “Any jitter...destroys communication and believability”
    - (quoting Disney animators Thomas and Johnston)
How about microsaccadic jitter?

• When we look at something:
  - our eyes are never perfectly still

• When others look at us:
  - perceptual system is sensitive to and amplifies small fluctuations

from Martinez-Conde Laboratory
Eye movement simulation

- Not modeling musculature or gimbaled movement
  - sophisticated models of extraocular muscles do exist
- Sufficient to model look point

Duchowski and Jörg (2015)
Look point simulation

* Note: no GPU required!
Look point simulation

Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. An eye tracker is a device for measuring eye positions and eye movement. Eye trackers are used in research on the visual system, in psychology, in psycholinguistics, marketing, as an input device for human-computer interaction, and in product design. There are a number of methods for measuring eye movement. The most popular variant uses video images from which the eye position is extracted. Other methods use search coils or are based on the electrooculogram.

Rayner (1998)

* Note: no GPU required!
Look point simulation

* Note: no GPU required!
Stochastic modeling

- Saccade duration via main sequence:
  \[ \Delta t = 2.2\mathcal{N}(\theta, \sigma = 10^\circ) + 21 + \mathcal{N}(0, 0.01) \]

- Fixation duration (ms): \( \mathcal{N}(250, 50) \) or data-driven

- Microsaccadic jitter at fixation:
  \[ \mathbf{p}_{t+h} = \mathbf{p}_t + \mathcal{P}(\alpha, \omega_0) \]
  - where \( \mathcal{P}(\alpha, \omega_0) \) is a pink noise filter with two parameters:
  - \( 1/f^\alpha \) describes power spectral distribution
  - \( \omega_0 \) is the filter’s unity gain frequency

Kretjtz et al. (2017)
Why $1/f^\alpha$ pink noise?

- Good for modeling physical/biological distributions
  - e.g., plants (Ostling et al., 2000) and galaxies (Landy, 1999)
  - behavior of biosystems in general (Szendro et al., 2001)

- Also good for modeling microsaccadic jitter
  - fixation error & neural noise combine to trigger microsaccades
  - can be approximated by $1/f^\alpha$ noise (Yang et al., 2009)

Zhou et al. (2012)
Why $1/f^\alpha$ pink noise?

- $1/f^\alpha$ neuronal noise accomplishes tradeoff:
  - memory inherent in $1/f^\alpha$ system serves as priming effect
  - sensitivity to small fluctuations (Usher et al., 1995)
Pink noise

![Graph showing different types of noise: White Noise, Pink Noise, Brown Noise](image)

- **White Noise**: Constant power across all frequencies.
- **Pink Noise**: Power decreases as frequency increases (rate of decrease is proportional to frequency).
- **Brown Noise**: Power decreases as frequency increases (rate of decrease is proportional to frequency squared).

**Power Spectral Density**

- $S(f) = k$
- $S(f) = 1/f$
- $S(f) = 1/f^2$

**Creating Noise**

*Stefan Hollas*

*J. Richard Hollas*
Pink noise

white noise

pink noise
Alternative approaches

- Self-avoiding random walk

Engbert 2012

ours (just starting out)
Alternative approaches
Stochastic modeling

- Saccade movement:  \( p_t = P_{i-1} + h(s)P_i \)
  - use force-time function (similar to limb movement)
  - assume symmetric-impulse model (Abrams et al., 1989)
  - combine two Hermite blending functions for \( \ddot{h}(s) \)

\[
\ddot{h}(s) = 2s^3 - 3s^2 + s \\
\dot{h}(s) = \frac{1}{2}s^4 - s^3 + \frac{1}{2}s^2 \\
h(s) = \frac{1}{10}s^5 - \frac{1}{4}s^4 + \frac{1}{6}s^3
\]
Microsaccadic jitter

no microsaccadic jitter

(too much?) microsaccadic jitter
Gaze-guided gristmill

- Gaze-guided gristmill
- fixations
- simulated microsaccades
- eye motion for rendering
- simulated tracker data
- event detection filter
- simulator
- real data
- eye tracker noise

- Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of the eyes. It is used in research on the visual system, in psychology for the study of attention or vision, and in ergonomics as a measure of user interaction, and its related devices. There are a number of methods for engineering and measurement. The most popular screen-based methods are those that rely on the end position to position. Other methods are based on the epipolar discontinuity.
Eye movement synthesis

Jörg et al. (SAP 2018)
Summary: Expressive eyes

- Outstanding problems:
  - need more expressive eyes (based on experiments)
  - periocular skin animation (eyelids, etc.)
  - caruncle
Eye Tracking

Research & Applications

- assistive
  - selection/look to shoot
  - translation/interpretation

- active
  - foveated rendering (gaze-contingent displays)

- passive
  - gaze synthesis

- expressive
  - assessment/training

- diagnostic (off-line)
Motivation: accessible 360 video in VR

- Predominant subtitling methods might not work well in VR
- Fixed, head-locked subs may not indicate location of speaker
- For example, person below in pic is *not* speaking
Background

• Various subtitling schemes have been tried by the BBC
  - evenly spaced, fixed below eye line
  - head-locked, always visible, fixed within view
  - head-locked with lag
  - appearing in front, then fixed

• Results somewhat inconclusive
  - various rendering options available
  - newer methods are needed to evaluate them

Hughes et al. (2015)
Objective: measure eye movements in VR

- Eye-tracking methodology is needed to test subtitling methods
- Methodology affords objective comparisons, not just subjective
- Metrics could lead to novel models of reading
Live Web Testing Framework

- Web-based subtitle testing framework ported to VR (Unity 3D)
Live Web Testing Framework

- 3 components: video, fake camera, and subtitle containers
Live Web Testing Framework

- Allows various subtitle rendering styles, various parameters
Live Web Testing Framework

- Allows recording and playback of gaze, subtitles, 360 video
Live Web Testing Framework

- Allows recording and playback of gaze, subtitles, 360 video
Objective: controlled empirical testing in VR

- Need to control 360 video stimulus, e.g., render / record video
- Either fully synthetic or principal photography
- Stimulus is scripted and directed like a short film
  - e.g., position actors / speakers at specific locations
Methodology: stimulus generation

- Script controls what is spoken and when (subtitle consistency)
- Principle photography recorded in 360 directly (Vuze 360 camera)
Methodology: apparatus

- 360 video projected in VR via Unity 3D
- Playback and gaze recording via HTC Vive Pro Eye
- Gaze sampling at 120 Hz in own thread, display at 90 Hz
Methodology: experimental design

• Experimental design is key to meaningful studies & results

• Ecological validity
  - push / pull between “in the wild” reality and “in the lab” control
  - controlled experiments are needed to establish baseline observations

• Preliminary study design: $2 \times 2 \times 2$ design with indep. variables:
  - stage (360 vs. 180)
  - masks (on vs. off)
  - object (human vs. abstract [cylinder])

• Idea is to compare how well subtitles are read (e.g., WPMs)
Methodology: gaze metrics

- Gaze signal can be analyzed in real-time or during playback
- Analysis relies on detecting fixations using velocity-based filter
- Further statistics rely on counting fixations in subtitles
Conclusion

- Architecture is ready for eye-tracked testing
- Next steps: conduct experiments in a controlled manner
- Analyze and report results
Eye Tracking

Research & Applications

assistive
active
selection/look to shoot
translation/interpretation

passive
foveated rendering (gaze-contingent displays)

expressive
gaze synthesis

diagnostic (off-line)
assessment/training
Task dependency

Ilya Efimovich Repin’s *An Unexpected Visitor* (Yarbus, 1967)
Task dependency

Ilya Efimovich Repin’s *Vsevolod Mikhailovich Garshin (1855-1888)*, 1884
Oil on canvas, Gift of Humanities Fund, Inc., 1972, The Metropolitan Museum of Art, New York, NY
Expert/novice assessment

Which is expert, which is novice?
Expert/novice assessment

Airway, Bones, Cardiac silhouette, Diaphragm, Extern tissue...
Training

Task: count immunogold (BrdU) stained epithelial cells in bovine mammary tissue

Gaze-Augmented Think-Aloud (Vitak et al., CHI 2012)
Training

• Effects on accuracy?
  - no effect on true positive rate (hits)
  - significant effect on false positive rate (error)

• GATA reduced making mistakes

Gaze-Augmented Think-Aloud (Vitak et al., CHI 2012)
Tracking experts

- Ericsson et al. (2006) surveyed experts’ gaze:
  - search strategies are task dependent
  - dwell times are shorter
  - make better use of extrafoveal/peripheral information
  - patterns of visual analysis develop with training
  - make use of larger visual span (area around fixation)

Ronaldo (Ergoneers), pilots (Weibel et al., ETRA 2012), laparoscopic surgeons (courtesy of Stella Atkins, Bin Zheng)
Serious applications

Peysakhovich (PhD Defense, 2016)
Serious applications

- Visual monitoring

Peysakhovich (PhD Defense, 2016)
Serious applications

• DR 400
Novice pilot (20 h) with instructor going over startup checklist.
Serious applications

• Edge bundling for visualization

Peysakhovich (PhD Defense, 2016)
Serious applications

- Flight / driving simulators
Serious applications

• Flight / driving simulators
Serious applications

- Flight / driving simulators

Expert

Novice
Serious applications

- Flight / driving simulators

Expert

Novice
Challenges: Areas Of Interest (AOIs)

- ArUco markers are popular
- OpenCV / OpenGL integration for visualization

Duchowski et al. (EMDPA 2020)
Challenges: Areas Of Interest (AOIs)

- Dynamic AOIs using SubRip format
Challenges: Areas Of Interest (AOIs)

- Dynamic AOIs using SubRip format
Challenges: Automatic AOI Segmentation

- Heuristics for eye, nose, mouth regions
Summary: Diagnostics

- Outstanding problems:
  - various analysis algorithms needed, e.g., comparison
  - visualization innovations

Ambient/focal visualization

Transition entropy

\[ H_t = - \sum_{i \in S} \pi_i \sum_{j \in S} \log p_{ij} \]
Summary: Diagnostics

Index of Pupillary Activity

\[ \dot{x}_n = \frac{x_{n+2} + x_{n+1} - x_{n-1} - x_{n-2}}{6\Delta t} \]
\[ \sigma_x = \sqrt{\langle \dot{x}^2 \rangle - \langle \dot{x} \rangle^2}, \quad \sigma_y = \sqrt{\langle \dot{y}^2 \rangle - \langle \dot{y} \rangle^2} \]
\[ \eta_x = \lambda \sigma_x, \quad \eta_y = \lambda \sigma_y \]
\[ x(t) = \sum_{j,k=-\infty}^{\infty} c_{j,k} \psi_{j,k}, \quad j, k \in \mathbb{Z} \]

- Outstanding problems:
  - measuring user state: microsaccades, pupil diameter
  - e.g., cognitive load

Duchowski et al. (CHI 2018)
For more information...

- Eye Tracking Methodology, 3rd ed. 2017
  - ISBN: 978-3-319-57881-1
  - over 4,000 citations

- Updated with:
  - advanced metrics
  - gaze analytics pipeline
  - and more...
For more information...

- Eye Tracking Methodology, **3rd ed.** 2017
  - ISBN: 978-3-319-57881-1
  - over 4,000 citations

- Updated with:
  - advanced metrics
  - gaze analytics pipeline
  - also available in Chinese
For more information...

[Image of book covers]
Eye Tracking

Future Directions

assistive
- subtitles
- training

active
- multi-modality
- gaze gestures

passive
- high-speed tracking
- saccade prediction

expressive
- expressiveness
- periocular skin animation

diagnostic (off-line)
- advanced analysis
- visualization
Eye Tracking

Gaze Analytics

Andrew T. Duchowski  duchowski@clemson.edu
Gaze Analytics Pipeline

Andrew T. Duchowski

Nina Gehrer
Gaze Analytics Pipeline: Origins

• What is it?
  - series of Python scripts followed by analysis in R
  - goal: automation

• How did it start, evolve?
  - ETH Winter School 2016
Gaze Analytics Pipeline: Ontology

• Where does it fit?
  - Note quite PyGaze (www.pygaze.org)
Gaze Analytics Pipeline: Ontology

- Where does it fit?
  - Note quite eyetrackingR (www.eyetracking-r.com)
Gaze Analytics Pipeline: Objectives

• How does it work?
  - key goals: visualization and analysis

see Gehrer et al., [2018], *Gaze Analytic Methods in Clinical Psychology*, ETRA 2018
Gaze Analytics Pipeline: Objectives

• How does it work?
  - key goals: visualization and analysis
  - like R’s tidyverse, sort of
  - idea is the same: import data, tidy, visualize, collate, analyze
  - each step a different Python script
Gaze Analytics Pipeline: Objectives

• How does it work?
  - key goals: visualization and analysis
Mock study on visibility of fiducial markers

- Modeled on art gallery display, e.g., mobile eye-tracking study
  - need to vary size and presence of maker (3 levels)
  - ideally cannot use same portrait more than once
Mock study on visibility of fiducial markers

- Modeled on art gallery display, e.g., mobile eye-tracking study
  - need to vary size and presence of maker (3 levels)
  - add another portrait (2 levels...or more)
Mock study on visibility of fiducial markers

- Stimulus presentation (PsychoPy)
  - shown for 4 s (arbitrary)
  - size of display and images matters (a lot!)
Apparatus

- EyeLink 1000 Eye Tracker (SR Research) or Gazepoint GP3
- Sampling at 500, 150, or 60 Hz
- Screen size: 22 inch
- Screen resolution: 1920 x 1080 pixels
- Screen distance: 60 cm
- 5-point calibration
Gaze Analytics Pipeline: Record

• How to record data (for offline analysis)?
  - can use vendor software (e.g., Gazepoint Analysis, EyeLink, ...)
Gaze Analytics Pipeline: Record

• How to record data (for offline analysis)?
  - can use vendor software (e.g., Gazepoint Analysis, EyeLink, ...)

![Gaze Analytics Pipeline Diagram]
Gaze Analytics Pipeline: Record

- How to record data (for offline analysis)?
  - can use vendor software (e.g., Gazepoint Analysis, EyeLink, ...)
  - can use free software: PsychoPy
Gaze Analytics Pipeline: Record

- PsychoPy and EyeLink, GazePoint, Tobii:
  - uses ioHub to communicate with eye trackers
  - saves data to HDF5 files
Gaze Analytics Pipeline: Record
Gaze Analytics Pipeline: Record

- PsychoPy and EyeLink, GazePoint, Tobii:
  - limited AOI functionality (but could be good, e.g., real-time)
  - need to code in Python various events, conditions (HDF5)
Gaze Analytics Pipeline: Record

- PsychoPy and HDF5:
  - many files and folders in one large file (.hdf5)
  - each data “file” stored as table (can be queried!)

<table>
<thead>
<tr>
<th>experiment_id</th>
<th>session_id</th>
<th>device_id</th>
<th>event_id</th>
<th>type</th>
<th>device_time</th>
<th>logged_time</th>
<th>time</th>
<th>confidence_interval</th>
<th>delay</th>
<th>filter_id</th>
<th>msg_offset</th>
<th>category</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>515</td>
<td>1</td>
<td>1419</td>
<td>151</td>
<td>95</td>
<td>95.008890</td>
<td>95.004850</td>
<td>95</td>
<td>95.00388</td>
<td>6.28E-4</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... Done Target Position Resul...</td>
<td></td>
</tr>
<tr>
<td>516</td>
<td>1</td>
<td>1420</td>
<td>151</td>
<td>95</td>
<td>95.004849</td>
<td>95.005415</td>
<td>95</td>
<td>95.0048</td>
<td>5.61E-4</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... reporting_unit_type.height</td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>1</td>
<td>1421</td>
<td>151</td>
<td>95</td>
<td>95.005802</td>
<td>95.006176</td>
<td>95</td>
<td>95.0098</td>
<td>3.74E-4</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... min_error 0.01492847415...</td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>1</td>
<td>1422</td>
<td>151</td>
<td>95</td>
<td>95.006461</td>
<td>95.006699</td>
<td>95</td>
<td>95.0064</td>
<td>4.57E00</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... max_error 0.006830000667...</td>
<td></td>
</tr>
<tr>
<td>519</td>
<td>1</td>
<td>1423</td>
<td>151</td>
<td>95</td>
<td>95.007186</td>
<td>95.007737</td>
<td>95</td>
<td>95.0071</td>
<td>5.51E-4</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... mean_error 0.0492829721...</td>
<td></td>
</tr>
<tr>
<td>520</td>
<td>1</td>
<td>1424</td>
<td>151</td>
<td>95</td>
<td>95.005017</td>
<td>95.008549</td>
<td>95</td>
<td>95.0080</td>
<td>5.32E-4</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... passed Test</td>
<td></td>
</tr>
<tr>
<td>521</td>
<td>1</td>
<td>1425</td>
<td>151</td>
<td>95</td>
<td>95.006889</td>
<td>95.006964</td>
<td>95</td>
<td>95.0098</td>
<td>7.04E-4</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... positions_failed_processin...</td>
<td></td>
</tr>
<tr>
<td>522</td>
<td>1</td>
<td>1426</td>
<td>151</td>
<td>95</td>
<td>95.010034</td>
<td>95.014391</td>
<td>95</td>
<td>95.0100</td>
<td>0.0043572</td>
<td>0</td>
<td>0</td>
<td>VALIDTA... Validation Report Complete</td>
<td></td>
</tr>
<tr>
<td>523</td>
<td>1</td>
<td>1490</td>
<td>151</td>
<td>97</td>
<td>97.681777</td>
<td>97.682218</td>
<td>97</td>
<td>97.6817</td>
<td>4.41E-4</td>
<td>0</td>
<td>0</td>
<td>trial start</td>
<td></td>
</tr>
<tr>
<td>524</td>
<td>1</td>
<td>2090</td>
<td>151</td>
<td>107</td>
<td>107.66264</td>
<td>107.68226</td>
<td>107</td>
<td>107.6822</td>
<td>6.24E-4</td>
<td>0</td>
<td>0</td>
<td>trial stop</td>
<td></td>
</tr>
</tbody>
</table>
Gaze Analytics Pipeline: Record

- PsychoPy and HDF5:
  - gaze data needs to be extracted between start/stop timestamps
  - timestamps need to be recorded in event table
Gaze Analytics Pipeline: Record

- PsychoPy and Scribus:
  - AOIs can be defined by yet another (free) program: Scribus
  - files saved as .sla (xml) files that can be parsed (e.g., Python)
Gaze Analytics Pipeline: Record

• Putting it all together: PsychoPy, HDF5, Scribus
  - all held together by Python (and R for statistics)
  - all multi-platform (macOS, Windows, Linux too [I think])
Data

http://andrewd.ces.clemson.edu/hcii23/demo.zip
Data

Participants: 24 students

<table>
<thead>
<tr>
<th>File Name</th>
<th>Date/Time</th>
<th>Type</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR_es01mb.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>15.982 KB</td>
</tr>
<tr>
<td>SOR_es02te.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.179 KB</td>
</tr>
<tr>
<td>SOR_es03lb.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.260 KB</td>
</tr>
<tr>
<td>SOR_es05sk.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>15.928 KB</td>
</tr>
<tr>
<td>SOR_es06ss.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.260 KB</td>
</tr>
<tr>
<td>SOR_es07sg.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.261 KB</td>
</tr>
<tr>
<td>SOR_es09ls.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.405 KB</td>
</tr>
<tr>
<td>SOR_es10aw.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>15.984 KB</td>
</tr>
<tr>
<td>SOR_es13lg.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.458 KB</td>
</tr>
<tr>
<td>SOR_es14mh.txt</td>
<td>18.01.2018 13:34</td>
<td>Textdokument</td>
<td>16.222 KB</td>
</tr>
</tbody>
</table>
Traditional and advanced gaze analytics

1) Measures related to Areas of Interest (AOIs)
   - Number of fixations, fixation durations
   - Frequency of the initial fixation after stimulus onset
   - Number of transition between AOIs

2) Measures of scanning behavior in general
   - IPA (Index of Pupillary Activity) / LHIPA (Low/High IPA)
   - K coefficient (ambient / focal fixations)
   - Microsaccades (rate, amplitude)
   - Transition matrices and transition entropy
Preparation for Analysis:

Structure of directories and files

- aois: AOI (Scribus) files
- exp: data, stimuli, and PsychoPy experiment
- src: Python, R source code

Definition of AOs in Scribus:
- painting
- largeQRcode
- location
- painter_tile
- smallQRcode
- acquisition

museum_signage.sla
Gaze analytics pipeline overview

- Main targets (e.g., *nix Makefile or Windows bat files)
- Idea is to type one command and go for coffee
- Return from coffee and write paper
- 5 easy steps
Gaze analytics pipeline overview

• Main targets (e.g., *nix Makefile or Windows bat files)
• Idea is to type one command and go for coffee
• Return from coffee and write paper
• 5 easy steps

1. dirs
Gaze analytics pipeline overview

- Main targets (e.g., *nix Makefile or Windows bat files)
- Idea is to type one command and go for coffee
- Return from coffee and write paper
- 5 easy steps:
  1. dirs; 2. raw
Gaze analytics pipeline overview

- Main targets (e.g., *nix Makefile or Windows bat files)
- Idea is to type one command and go for coffee
- Return from coffee and write paper

5 easy steps:
1. dirs; 2. raw; 3. process
Gaze analytics pipeline overview

- Main targets (e.g., *nix Makefile or Windows bat files)
- Idea is to type one command and go for coffee
- Return from coffee and write paper
- 5 easy steps:
  1. dirs; 2. raw; 3. process; 4. collate
Gaze analytics pipeline overview

- Main targets (e.g., *nix Makefile or Windows bat files)
- Idea is to type one command and go for coffee
- Return from coffee and write paper
- 5 easy steps:
  1. dirs; 2. raw; 3. process; 4. collate; 5. stats
Gaze analytics pipeline overview

The python and R scripts:

1. mkdir
   set up directory (basic OS command)
2. hdf52raw.py
   parse vendor data into .raw data
3. filter.py
   process .raw data (event detection)
   graph.py
   visualize data
4. collate-* .py
   collate to .csv data
5. *.R
   do the stats

Linux or macOS: use Makefile
Windows: use .bat files
Gaze analytics pipeline overview

Windows (using .bat files):

1. .\dirs.bat set up directory (basic OS command)
2. .\raw.bat parse vendor data into .raw data
3. .\process.bat process .raw data (event detection)
   .\graph.bat visualize data
4. .\collate.bat collate to .csv data
5. .\stats.bat do the stats

Windows: or use .\make.bat file
Gaze analytics pipeline overview

Linux or macOS (using Makefile):

1. make dirs  set up directory
2. make raw   parse vendor data into .raw data
3. make process  process .raw data (event detection)
   make graph  visualize data
4. make collate  collate to .csv data
5. make stats   do the stats

Linux or macOS:  or simply use make
Gaze analytics pipeline: essential information

- All of this information is used by scripts:
  - screen resolution: 1920 x 1080
  - screen dimensions (diagonal): 22 in
  - sampling rate: 60 Hz
  - viewing distance: 21.65 in (55 cm)
  - AOIs: software (Scribus)

- Also need directories (all paths relative!):
  - indir: ../exp/data/
  - imgdir: ../exp/stimuli/static/screenshots
  - pltdir: ./plots/
  - outdir: ./data
  - rawdir: ./data/raw
Gaze analytics pipeline: objectives

- Process raw gaze data into fixations, fixation count, etc.

<table>
<thead>
<tr>
<th>visual angle conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>- width, height of screen (e.g., 1920 x 1080)</td>
</tr>
<tr>
<td>- screen dimensions (diagonal, e.g., 22 inches)</td>
</tr>
<tr>
<td>- viewing distance (e.g., 21.65 inches)</td>
</tr>
</tbody>
</table>
# Gaze analytics pipeline: objectives

- Process raw gaze data into fixations, fixation count, etc.

<table>
<thead>
<tr>
<th>visual angle conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>• width, height of screen (e.g., 1920 x 1080)</td>
</tr>
<tr>
<td>• screen dimensions (diagonal, e.g., 22 inches)</td>
</tr>
<tr>
<td>• viewing distance (e.g., 21.65 inches)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Butterworth smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• filter order (e.g., 2nd, 4th, etc.)</td>
</tr>
<tr>
<td>• sampling rate (e.g., 60 Hz)</td>
</tr>
<tr>
<td>• cutoff frequency (e.g., 6.15 Hz)</td>
</tr>
</tbody>
</table>
Gaze analytics pipeline: objectives

- Process raw gaze data into fixations, fixation count, etc.

<table>
<thead>
<tr>
<th>visual angle conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>- width, height of screen (e.g., 1920 x 1080)</td>
</tr>
<tr>
<td>- screen dimensions (diagonal, e.g., 22 inches)</td>
</tr>
<tr>
<td>- viewing distance (e.g., 21.65 inches)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Butterworth smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>- filter order (e.g., 2nd, 4th, etc.)</td>
</tr>
<tr>
<td>- sampling rate (e.g., 60 Hz)</td>
</tr>
<tr>
<td>- cutoff frequency (e.g., 6.15 Hz)</td>
</tr>
</tbody>
</table>
Gaze analytics pipeline: objectives

- Process raw gaze data into fixations, fixation count, etc.

<table>
<thead>
<tr>
<th>visual angle conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>- width, height of screen (e.g., 1920 x 1080)</td>
</tr>
<tr>
<td>- screen dimensions (diagonal, e.g., 22 inches)</td>
</tr>
<tr>
<td>- viewing distance (e.g., 21.65 inches)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Butterworth smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>- filter order (e.g., 2nd, 4th, etc.)</td>
</tr>
<tr>
<td>- sampling rate (e.g., 60 Hz)</td>
</tr>
<tr>
<td>- cutoff frequency (e.g., 6.15 Hz)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Savitzky-Golay differentiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>- filter width (e.g., 3)</td>
</tr>
<tr>
<td>- degree (e.g., 3, 3rd order)</td>
</tr>
<tr>
<td>- order (e.g., 1 for differentiation, 0 for smoothing)</td>
</tr>
</tbody>
</table>
Gaze analytics pipeline: objectives

- Process raw gaze data into fixations, fixation count, etc.

<table>
<thead>
<tr>
<th>visual angle conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>width, height of screen (e.g., 1920 x 1080)</td>
</tr>
<tr>
<td>screen dimensions (diagonal, e.g., 22 inches)</td>
</tr>
<tr>
<td>viewing distance (e.g., 21.65 inches)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Butterworth smoothing</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter order (e.g., 2nd, 4th, etc.)</td>
</tr>
<tr>
<td>sampling rate (e.g., 60 Hz)</td>
</tr>
<tr>
<td>cutoff frequency (e.g., 6.15 Hz)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Savitzky-Golay differentiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter width (e.g., 3)</td>
</tr>
<tr>
<td>degree (e.g., 3, 3rd order)</td>
</tr>
<tr>
<td>order (e.g., 1 for differentiation, 0 for smoothing)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>thresholding</th>
</tr>
</thead>
<tbody>
<tr>
<td>velocity (e.g., 36 deg/s)</td>
</tr>
</tbody>
</table>
Critical notes on scripts

• None of the scripts are ready “out of the box”
• None of the scripts can easily be ported to other projects

• Why? Not possible to predict future study design

• What needs to be adapted?
  - file name composition, e.g.,
    s01-1-1-absent-female.raw
  - file name encodes (and can accommodate many more parameters):
    subj-exp_id-ses_id-marker-object
Gaze analytics pipeline: vendor data

- Vendor data comes in various formats, usually plain text

<table>
<thead>
<tr>
<th>RECORDING_SESSION_LABEL</th>
<th>Group_VP</th>
<th>blocknr</th>
<th>trialnr</th>
<th>task</th>
<th>emotion</th>
<th>gender</th>
<th>face</th>
<th>dc_x</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>TIMESTAMP</td>
<td>LEFT_GAZE_X</td>
<td>LEFT_GAZE_Y</td>
<td>LEFT_PUPIL_SIZE</td>
<td>RIGHT_GAZE_X</td>
<td>RIGHT_GAZE_Y</td>
<td>RIGHT_PUPIL_SIZE</td>
<td>RESPONSE</td>
</tr>
<tr>
<td>es01mb stud</td>
<td>1</td>
<td>1</td>
<td>emotion</td>
<td>surprised</td>
<td>male</td>
<td>M08</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>2154988,00</td>
<td>78,10</td>
<td>463,20</td>
<td>1296,00</td>
<td>126,00</td>
<td>467,40</td>
<td>1205,00</td>
<td>surprised</td>
<td>Correct</td>
</tr>
<tr>
<td>es01mb stud</td>
<td>1</td>
<td>1</td>
<td>emotion</td>
<td>surprised</td>
<td>male</td>
<td>M08</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>2154990,00</td>
<td>77,80</td>
<td>464,10</td>
<td>1299,00</td>
<td>126,30</td>
<td>467,20</td>
<td>1204,00</td>
<td>surprised</td>
<td>Correct</td>
</tr>
<tr>
<td>es01mb stud</td>
<td>1</td>
<td>1</td>
<td>emotion</td>
<td>surprised</td>
<td>male</td>
<td>M08</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>2154992,00</td>
<td>79,30</td>
<td>464,50</td>
<td>1296,00</td>
<td>125,50</td>
<td>467,00</td>
<td>1202,00</td>
<td>surprised</td>
<td>Correct</td>
</tr>
<tr>
<td>es01mb stud</td>
<td>1</td>
<td>1</td>
<td>emotion</td>
<td>surprised</td>
<td>male</td>
<td>M08</td>
<td>80</td>
<td>1</td>
</tr>
<tr>
<td>2154994,00</td>
<td>80,50</td>
<td>464,50</td>
<td>1294,00</td>
<td>125,00</td>
<td>467,00</td>
<td>1201,00</td>
<td>surprised</td>
<td>Correct</td>
</tr>
</tbody>
</table>

...
Gaze analytics pipeline: vendor data

- Vendor data comes in various formats, usually plain text
- Or HDF5 (organized as tables or data frames)
Gaze analytics pipeline: parse vendor data

• Just want to extract raw (unprocessed) data: $(x, y, t)$

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$t$</th>
<th>$t_id$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.079785</td>
<td>0.613216</td>
<td>61.449674</td>
<td>2235252.000000</td>
</tr>
<tr>
<td>0.079443</td>
<td>0.612891</td>
<td>61.602661</td>
<td>2235254.000000</td>
</tr>
<tr>
<td>0.079785</td>
<td>0.612630</td>
<td>61.602661</td>
<td>2235256.000000</td>
</tr>
<tr>
<td>0.079980</td>
<td>0.613411</td>
<td>61.398678</td>
<td>2235258.000000</td>
</tr>
<tr>
<td>0.080029</td>
<td>0.613802</td>
<td>61.398678</td>
<td>2235260.000000</td>
</tr>
<tr>
<td>0.080029</td>
<td>0.613997</td>
<td>61.398678</td>
<td>2235262.000000</td>
</tr>
<tr>
<td>0.079736</td>
<td>0.613932</td>
<td>61.398678</td>
<td>2235264.000000</td>
</tr>
<tr>
<td>0.079443</td>
<td>0.613411</td>
<td>61.398678</td>
<td>2235266.000000</td>
</tr>
<tr>
<td>0.079541</td>
<td>0.613086</td>
<td>61.449674</td>
<td>2235268.000000</td>
</tr>
<tr>
<td>0.079541</td>
<td>0.612956</td>
<td>61.347683</td>
<td>2235270.000000</td>
</tr>
<tr>
<td>0.079395</td>
<td>0.612760</td>
<td>61.245692</td>
<td>2235272.000000</td>
</tr>
<tr>
<td>0.079297</td>
<td>0.612630</td>
<td>61.398678</td>
<td>2235274.000000</td>
</tr>
<tr>
<td>0.079932</td>
<td>0.612370</td>
<td>61.602661</td>
<td>2235276.000000</td>
</tr>
<tr>
<td>0.080127</td>
<td>0.612500</td>
<td>61.500669</td>
<td>2235278.000000</td>
</tr>
<tr>
<td>0.079541</td>
<td>0.612891</td>
<td>61.500669</td>
<td>2235280.000000</td>
</tr>
<tr>
<td>0.079541</td>
<td>0.612760</td>
<td>61.551665</td>
<td>2235282.000000</td>
</tr>
<tr>
<td>0.079590</td>
<td>0.613411</td>
<td>61.551665</td>
<td>2235284.000000</td>
</tr>
</tbody>
</table>

...
Gaze analytics pipeline: graph

• Check stimulus image and AOI position
• Important to verify coordinates

*-aois.pdf
Gaze analytics pipeline: graph

- Check raw data (2D)

\[ g_i = (x_i, y_i, t_i) \]
Gaze analytics pipeline: graph

- Check fixations in AOIs

\[ \dot{x}_n^s(t) = \frac{1}{\Delta t^s} \left( \sum_{i=-p}^{p} h_i^t s x_{n-i} - \sum_{i=-q}^{q} g_i^t s \dot{x}_{n-i} \right) \]

- Savitzy-Golay filter
Gaze analytics pipeline: graph

• Check fixations in AOIs

\[
\dot{x}_{n}^{s}(t) = \frac{1}{(\Delta t^{s})} \left( \sum_{i=-p}^{p} h_{i}^{t,s} x_{n-i}^{t,s} - \sum_{i=-q}^{q} g_{i}^{t,s} \dot{x}_{n-i}^{t,s} \right)
\]

* -fxtn-aoi.pdf

- Savitzy-Golay filter

AOI Fixations
Gaze analytics pipeline: graph

- Check ambient/focal fixations

- fixation dur. – sacc. ampl.
- z-scores
- $\kappa > 0$ focal viewing
- $\kappa < 0$ ambient viewing

\[
\kappa_i = \frac{d_i - \mu_d}{\sigma_d} - \frac{a_{i+1} - \mu_a}{\sigma_a}
\]
Gaze analytics pipeline: graph

- Check microsaccades within ambient/focal fixations

\[ \dot{x}_n = \frac{x_{n+2} + x_{n+1} - x_{n-1} - x_{n-2}}{6\Delta t} \]

\[ \sigma_x = \sqrt{\langle \dot{x}^2 \rangle - \langle \dot{x} \rangle^2}, \quad \sigma_y = \sqrt{\langle \dot{y}^2 \rangle - \langle \dot{y} \rangle^2} \]

\[ \eta_x = \lambda \sigma_x, \quad \eta_y = \lambda \sigma_y \]
Gaze analytics pipeline: graph

- Can do this over grid / calibration image (validation)
- Did you remember to include this in the stimuli?
Gaze analytics pipeline: graph

- This really helps in fine-tuning event detection filters
- Can also compute your own accuracy & precision
  - really useful for reporting

\[
A = \sum_{i=1}^{M} \left( \sum_{j=1}^{N} \frac{\|T_i - P_{i,j}\|}{N} \right) \frac{1}{M}
\]
Gaze analytics pipeline: graph

- Look at microsaccades in ambient/focal fixations again
- Cause they’re cool
• Once happy with visualizations, process data
• Will end up with various *.dat files, one per subject:

1. *-pdwt.dat  wavelet transform (nothing to see here)
2. *-pICA.dat  Index of Pupillary Activity (IPA)
3. *-pups.dat  pupil diameter data (tricky)
4. *-smth.dat  smoothed (Butterworth) data
5. *-fxtn.dat  fixations
6. *-sacc.dat  saccades
7. *-msac.dat  microsaccades
8. *-msrt.dat  microsaccade rate
9. *-amfo.dat  ambient/focal K coefficient
10. *-fxtn-aois.dat  fixations in AOIs
Gaze analytics pipeline: collate

- Now collate data to prepare for stats processing
- Will end up with various *.csv files, one per metric:

1. pdwt.csv  wavelet transform (wouldn’t make sense)
2. pICA.csv  Index of Pupillary Activity (IPA)
3. pups.csv  pupil diameter (can get this, need baseline)
4. smth.csv  smoothed (Butterworth) data
5. fxtn.csv  fixations
6. sacc.csv  saccades
7. msac.csv  microsaccades
8. msrt.csv  microsaccade rate
9. amfo.csv  ambient/focal K coefficient
10. fxtn-aois.csv  fixations in AOIs
Gaze analytics pipeline: analyze

- Traditional metrics
  - fixations
  - fixation durations

- Novel / advanced metrics
  - ambient / focal fixations
  - Index of Pupillary Activity
  - microsaccade amplitude, rate

Discerning Ambient/Focal Attention with Coefficient K

SEZAYIZOF KLEITZ, National Information Processing Institute, Warsaw, Poland and University of Social Science and Humanities, Warsaw, Poland
ANDREW DOUCET, University of Cergy-Pontoise, Cergy, FRANCE
ZIAD L. E. WARD, University of Social Science and Humanities, Warsaw, Poland
AGNIESKA KALISZKA, National Information Processing Institute, Warsaw, Poland
AGATA R. POPP, National Information Processing Institute, Warsaw, Poland

The authors acknowledge K. diez, who devised the novel method for measuring eye movements. This method is based on the diez algorithm and features a novel measure of visual attention. The diez algorithm is designed to detect areas of high visual attention and to track changes in eye movement patterns. This novel approach may be useful in applications such as visual search, reading, and visual attention in virtual environments.

1. INTRODUCTION

There is an increasing demand for characterization of visual behavior through analysis of eye movements. Efforts to analyze eye movements have traditionally focused on traditional metrics such as fixations and fixation durations. In contrast, recent advances in eye tracking technology have enabled the measurement of novel metrics, such as ambient and focal fixations, and the index of pupillary activity (IPA). These metrics provide a more nuanced understanding of visual behavior and can be used to better characterize attentional processes.

This work has been partly supported by a grant from the National Information Processing Institute, Warsaw, Poland.
Gaze analytics pipeline: analyze

- Traditional metrics
  - fixations
  - fixation durations

- Novel / advanced metrics
  - ambient / focal fixations
  - Index of Pupillary Activity
    - microsaccade amplitude, rate

The Index of Pupillary Activity

Measuring Cognitive Load through Task Difficulty with Pupil Oscillation

Andrew T. Duchowski
School of Computing
Clemson University
Andrew.Duchowski@Clemson.edu

Cezary Walek & Anna Niedzielska
Interactive Technologies Laboratory
GdFIR
(cezary.walek@intercam.gdansk.pl)

Peter Kiefer, Martin Raubal & Ioannis Giammarino
Institute of Cognitive and Information Theory ETH Zurich
Peter.Kiefer@inf.ethz.ch, Ioannis.Giammarino@inf.ethz.ch

ABSTRACT
A novel measure of the frequency and duration of pupil dilation is proposed for evaluating subjective arousal levels and cognitive strain. The proposed novel measure, named the Index of Pupillary Activity (IPA), is obtained by subtracting a polynomial function of pupil size from pupil size metrics that can be assessed in an experiment where participants perform a mental arithmetic task while being exposed to varying levels of cognitive load. The polynomial function is derived from the relationship between pupil size and cognitive load, which is determined in a separate experiment. The IPA is further evaluated as a cognitive load measure in a controlled experiment with the goal of comparing its performance to traditional measures.

Keywords: pupilometry, cognitive task difficulty

ACM Classification Keywords
- H.5.1. Information interfaces and presentation: User-centered design

INTRODUCTION
Systems that can detect and respond to users' cognitive loads provide important new opportunities to enhance productivity and reduce cognitive strain. Such systems may benefit from reliable and fast methods that allow subjective cognitive strain to be measured and monitored online. Pupil dilation is an established measure of cognitive load, with higher levels of pupil dilation corresponding to higher levels of cognitive load (e.g., [1, 2, 3, 4]). This technology is already in use in applications for driving assistance, visual attention, and other domains. In the current study, we present a novel measure, the Index of Pupillary Activity (IPA), which provides a more accurate and flexible measure of cognitive load compared to traditional methods. The IPA is calculated by subtracting a polynomial function from pupil size metrics, which is determined in a separate experiment. The IPA is further evaluated as a cognitive load measure in a controlled experiment with the goal of comparing its performance to traditional measures.

The IPA is calculated as follows:

\[ IPA(t) = P(t) - \frac{a}{(t-b)^n} \]

where \( P(t) \) is the pupil size at time \( t \), \( a \), \( b \), and \( n \) are parameters that are determined in the separate experiment, and \( n \) is a positive integer.

The IPA is further evaluated in a controlled experiment where participants perform a mental arithmetic task while being exposed to varying levels of cognitive load. The IPA is compared to traditional measures of cognitive load, such as pupil size metrics, to determine its effectiveness as a cognitive load measure.

The results show that the IPA is a more accurate and flexible measure of cognitive load compared to traditional methods. The IPA is further evaluated in a controlled experiment with the goal of comparing its performance to traditional measures.
Gaze analytics pipeline: analyze

- Traditional metrics
  - fixations
  - fixation durations

- Novel / advanced metrics
  - ambient / focal fixations
  - Index of Pupillary Activity
  - microsaccade amplitude, rate
Gaze analytics pipeline: analyze

• Traditional metrics
  - fixations
  - fixation durations

• Novel / advanced metrics
  - ambient / focal fixations
  - Index of Pupillary Activity
  - microsaccade amplitude, rate

Abstract

Pupil diameter and microsaccades are captured by an eye tracker and compared for their suitability as indicators of cognitive load (as measured by task difficulty). Specifically, new metrics are tested in response to task difficulty: (1) the change in pupil diameter with respect to inter- or intra-trial baselines, and (2) the rate and magnitude of microsaccades. Participants performed easy and difficult mental arithmetic tasks while leading a central target. Inter-trial change in pupil diameter and microsaccade magnitude appears to adequately discriminate task difficulty, and hence cognitive load. If this implied causality can be assessed, this paper’s contribution combines previous work concerning microsaccade magnitude and extends this work by directly comparing microsaccade magnitudes to pupillometric measures. To our knowledge this is the first study to compare the reliability and sensitivity of task-related pupil diameter and microsaccade measures of cognitive load.

1 Introduction

Cognitive Load Theory (CLT) [1] plays an important role in Human-Computer Interaction (HCI) research. There is a pressing need for more precise measures of individuals’ cognitive load, as it can guide designers of interactive systems to avoid overwhelming users. Measurement of cognitive load could allow a return to expected appropriateness, potentially either by titrating doses or raising or lowering the level of task difficulty e.g., on off-learning systems [2], or by adapting missions critical to the user’s cognitive state [3]. Examples of the use include a wide range of applications, including surgery [4], flight control [5], human-robot design, human cognition modeling, usability, and multimedia learning [5]. A reliable, and possibly real-time, measurement of cognitive load is thus highly desirable. However, due to a lack of reliable measures, only a small body exists between Human-Computer Interaction and Cognitive Load Theory [5].

Yildiz et al. [2] for the present measurement methods in CLT studies are self-reporting, the dual task paradigm, and physiological measures (see also [4]). Eye tracking, a type of physiological measurement, may offer the greatest potential for a reliable, non-invasive
Gaze analytics pipeline: analyze

• Traditional metrics
  - fixations
  - fixation durations

• Novel / advanced metrics
  - ambient / focal fixations
  - Index of Pupillary Activity
  - microsaccade amplitude, rate

\[
\dot{x}_n = \frac{x_{n+2} + x_{n+1} - x_{n-1} - x_{n-2}}{6\Delta t}
\]

\[
\sigma_x = \sqrt{\langle \dot{x}^2 \rangle - \langle \dot{x} \rangle^2}, \quad \sigma_y = \sqrt{\langle \dot{y}^2 \rangle - \langle \dot{y} \rangle^2}
\]

\[
\eta_x = \lambda \sigma_x, \quad \eta_y = \lambda \sigma_y
\]
Gaze analytics pipeline: read

- Eye Tracking Methodology, *3rd ed.* 2017
  - ISBN: 978-3-319-57881-1

- More details found in book:
  - additional metrics (NNI)
  - microsaccades
  - heatmap visualization
  - binocular eye movement analysis
  - etc.
Gaze analytics pipeline: statistics

• Excellent online references for R (maybe dated by now):
  - Baron and Li’s Notes on the use of R for psychology experiments and questionnaires:
    • http://www.psych.upenn.edu/~baron/rpsych.html
  - The Personality Project
    • http://www.personality-project.org/r
Gaze analytics pipeline overview

- Main targets (e.g., *nix Makefile or Windows bat files)
- Idea is to type one command and go for coffee
- Return from coffee and write paper
- 5 easy steps:

1. dirs; 2. raw; 3. process; 4. collate; 5. stats
Gaze analytics pipeline:

*.csv files, one per metric:

amfo.csv ambient/focal K coefficient
fxtn.csv fixations
fxtn-aois.csv fixations in AOIs
msac.csv microsaccades
msrt.csv microsaccade rate
pICA.csv Index of Pupillary Activity (IPA)
pICALH.csv Low/High Index of Pupillary Activity (LHIPA)
sacc.csv saccades
Gaze analytics pipeline:

**One row per fixation:**
- `amfo.csv` ambient/focal K coefficient
- `ftxn.csv` fixations
- `msrt.csv` microsaccade rate

**One row per fixation in one of the AOIs:**
- `ftxn-aois.csv` fixations in AOIs

**One row per microsaccade**
- `msac.csv` microsaccades

**One row per trial (per subject)**
- `pICA.csv` Index of Pupillary Activity (IPA)
- `pICALH.csv` Low/High Index of Pupillary Activity (IPA)

**One row per saccade:**
- `sacc.csv` saccades
Gaze analytics pipeline: statistics

1. fxtn.R  
   gaze analytics related to fixations, AOIs

2. tm.R  
   transition matrices and transition entropy

For all others, write your own
Gaze analytics pipeline: statistics
Gaze analytics pipeline: statistics

1. fxtn.R  
gaze analytics related to fixations, AOIs

2. tm.R  
transition matrices and transition entropy

Will end up with figures in ./figs and 4 *.out files, one per R script:

fxtn.out
tm.out
Gaze analytics pipeline: statistics

Code and output/statistics in the *.out files:

fxtn.out

```r
> # Fixation duration.................................
> (fit <- aovتز\(duration\) ~ primitive,  
+ id = "subject", 
+ dfv = "duration", 
+ within = c('marker', 'object'), 
+ type = 3, 
+ factorize = FALSE))

Anova Table (Type 3 tests)

Response: duration

             Effect     df  MSE    F  gas p.value
 1 marker 1.99, 5.96 0.00 0.04 0.004 .962
 2 object 1, 3 0.00 2.32 .019 .225
 3 marker:object 1.48, 4.43 0.00 0.36 0.028 .657

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Sphericity correction method: GG
> summary(fit)

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

                          Sum Sq  num Df Error SS  den Df F value  Pr(>F)
(Intercept)            0.55046     1  0.042975  3  38.4250  0.000048 **
marker         0.00033     2  0.025532  6  0.0387  0.952248
object      0.00075      1  0.002271  3  2.3171  0.225323
marker:object 0.00062     2  0.021759  6  0.3615  0.710742

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

calculated from ftxn.csv
Gaze analytics pipeline: statistics

Code and output/statistics in the *.out files:

```r
# Calculated from fxtn-aois.csv

# Code snippet

# Calculated from fxtn-aois.csv
```

<table>
<thead>
<tr>
<th>subj</th>
<th>exp_id</th>
<th>ses_id</th>
<th>marker</th>
<th>object</th>
<th>timestamp</th>
<th>x</th>
<th>y</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>0.048479</td>
<td>1373.0</td>
<td>0.016120</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>0.080980</td>
<td>1365.1017</td>
<td>0.177426</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>0.274906</td>
<td>1358.2080</td>
<td>0.015980</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>0.388075</td>
<td>1253.0232</td>
<td>0.016155</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>0.428441</td>
<td>1223.0428</td>
<td>0.388319</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>0.894711</td>
<td>1185.3912</td>
<td>0.194814</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>1.098834</td>
<td>1180.5486</td>
<td>0.193978</td>
<td></td>
</tr>
<tr>
<td>s03</td>
<td>1</td>
<td>1</td>
<td>large</td>
<td>male</td>
<td>1.196911</td>
<td>1334.3032</td>
<td>0.016082</td>
<td></td>
</tr>
</tbody>
</table>
Effect of markers on fixation duration

1) Measures related to Areas of Interest (AOIs) such as the marker and object (painting):

AOIs:
- object (painting),
- large marker, small marker,
- painter info,
- acquisition info
Effect of markers on fixation duration

1) Measures related to Areas of Interest (AOIs) such as the marker and object (painting):
e.g., fixation duration

Hypothesis (coincides with null hypothesis):
no difference expected (because markers are not distracting)

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MSE</th>
<th>F</th>
<th>ges</th>
<th>p.value</th>
</tr>
</thead>
<tbody>
<tr>
<td>marker</td>
<td>1.06, 3.19</td>
<td>4870.47</td>
<td>0.24</td>
<td>.020</td>
<td>.671</td>
</tr>
<tr>
<td>object</td>
<td>1, 3</td>
<td>2498.45</td>
<td>0.67</td>
<td>.027</td>
<td>.474</td>
</tr>
<tr>
<td>marker:object</td>
<td>1.44, 4.31</td>
<td>2400.13</td>
<td>1.22</td>
<td>.065</td>
<td>.356</td>
</tr>
</tbody>
</table>

Mean fixation duration (sec.)

- Absent
- Large Marker
- Small Marker
Effect of markers on gaze transition

1) Measures related to Areas of Interest (AOIs) such as the marker and object (painting):
e.g., gaze transition

Hypothesis (coincides with null hypothesis):
no difference expected (because markers are not distracting)

Need ANOVA on entropy between conditions (exercise for the reader)
Gaze analytics pipeline: transition entropy

- Normalized transition entropy
- Higher entropy means “surprise!”

\[ H_t = -\frac{1}{\log_2 s} \sum_{i \in S} \pi_i \sum_{j \in S} p_{ij} \log_2 p_{ij} \]
Gaze analytics pipeline: transition entropy

- Normalized transition entropy
- Higher entropy means “surprise!”
- Stationary entropy: long run

\[ H_s = - \sum_{i \in S} \pi_i \log \pi_i \]
Stationary entropy: example

- Predicting the weather:
  - if rainy, probability it remains rainy is .51
  - if sunny, 21% chance it become cloudy

\[ P = \begin{pmatrix}
  \text{rainy} & \text{cloudy} & \text{sunny} \\
  0.51 & 0.39 & 0.10 \\
  0.32 & 0.35 & 0.33 \\
  0.15 & 0.21 & 0.64 \\
\end{pmatrix} \quad \sum p_i = 1 \]
Stationary entropy: example

• Predicting the weather:
  - if rainy, probability it remains rainy is .51
  - if sunny, 21% chance it become cloudy

\[
P = \begin{pmatrix}
\text{rainy} & 0.51 & 0.39 & 0.10 \\
\text{cloudy} & 0.32 & 0.35 & 0.33 \\
\text{sunny} & 0.15 & 0.21 & 0.64 \\
\end{pmatrix}
\]

\[\sum p_i = 1\]
Stationary entropy: example

- Predicting the weather:
  - if rainy, probability it remains rainy is .51
  - if sunny, 21% chance it become cloudy

\[
P = \begin{pmatrix}
\text{rainy} & \text{cloudy} & \text{sunny} \\
0.51 & 0.39 & 0.10 \\
0.32 & 0.35 & 0.33 \\
0.15 & 0.21 & 0.64
\end{pmatrix} \quad \sum p_i = 1
\]
Predicting the weather:
- if rainy, probability it remains rainy is .51
- if sunny, 21% chance it become cloudy

Transition matrix gives likelihood of next period

\[
P = \begin{pmatrix}
0.51 & 0.39 & 0.10 \\
0.32 & 0.35 & 0.33 \\
0.15 & 0.21 & 0.64
\end{pmatrix}
\]

\[\sum p_i = 1\]

Stationary entropy: example

What about next two periods?
Stationary entropy: example

• What about next two periods? $PP = P_2$
  - if it’s rainy today, chance it will be sunny in 2 days is .25

$$P_2 = \begin{pmatrix} 0.51 & 0.39 & 0.10 \\ 0.32 & 0.35 & 0.33 \\ 0.15 & 0.21 & 0.64 \end{pmatrix} \begin{pmatrix} 0.51 & 0.39 & 0.10 \\ 0.32 & 0.35 & 0.33 \\ 0.15 & 0.21 & 0.64 \end{pmatrix} = \begin{pmatrix} 0.40 & 0.36 & 0.25 \\ 0.32 & 0.32 & 0.36 \\ 0.24 & 0.27 & 0.49 \end{pmatrix}$$

• Similarly, probability in the long is $P_n = P^n$
Stationary entropy: example

- In the long run $P_n = P^n$
- steady-state (stationary) transition probabilities converge
- steady-state vector $\pi$ is eigenvector of $P$ with eigenvalue $\lambda = 1$

\[ xP^n \rightarrow \pi \quad \forall x \quad \text{if} \quad \pi P = \lambda \pi \]
Stationary entropy: transition vs. stationary entropy?

- Ultimately, not super certain of stationary entropy’s utility

- Because: \[ H_t = -\frac{1}{\log_2 s} \sum_{i \in S} \pi_i \sum_{j \in S} p_{ij} \log_2 p_{ij} \quad H_s = -\sum_{i \in S} \pi_i \log \pi_i \]

  transition entropy is always smaller \( H_t \leq H_s \)

- Long-term distribution of transitions is expected to be more uniform
Gaze analytics pipeline: where to go from here?

- Important to remember what the pipeline offers: metrics
- Which metrics to use will depend on study hypothesis

- General strategy “recipe” for controlled experiments:
  - formulate hypothesis
    - don’t start with “I wonder what would happen if...”
    - start with “I bet this would happen if...”
  - design experiment (e.g., within-, between-subjects)
  - choose metrics
    - gaze metrics (process metrics) often supplement performance metrics
  - choose analytical tools (stats, e.g., ANOVA, something else)

- Can do exploratory research or pilot studies beforehand
Gaze analytics pipeline: write paper

• Remember analytics pipeline is meant to help automate analysis

• Once that’s done, write the paper

• This too has a basic “recipe”:
  - abstract, intro, background
  - hypothesis
  - recent trend is to register this a priori
  - methodology
  - design, stimulus, apparatus, procedure, participants
  - results
  - discussion
  - conclusions
Eye Tracking
Gaze Interaction
Gaze Interaction

• Real-time gaze input:
  - usually obtained from server via SDK / API
  - hardware driver may need to be running
  - e.g., Gazepoint Control
  - stream raw data $(x, y, t)$
Calibration

- Use vendor-provide or write your own
  - either way, collect error, keep track of accuracy
The Gazepoint eye tracker

- OpenGaze API (protocol)

- Key features:
  - no libraries/DLLs required
  - language-independent
  - standard TCP/IP socket
  - XML “packets”
How to start?

• Gazepoint’s API
  - not really an API, a protocol, really

• Eye tracking server
  - controls eye tracking device
  - streams out data

• Client app (e.g., Unity 3D)
  - connects (TCP/IP) to server
Example client/server architecture

- From Gazepoint:
  - no SDK
  - you write networking code
  - send/receive XML packets
  - over network or to 127.0.0.1
Eye tracker communication

• Generally, communication with an eye tracker entails:
  - network connection, e.g., open TCP/IP socket
  - can be embodied in an object
  - usually a thread is created to accept streaming data

```python
class Gazept:
    def __init__(self, server="127.0.0.1", port=4242):
        self.server = server
        self.port = port
        try:
            self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            self.sock.connect((server, port))
            self.receiving_thread = threading.Thread(target=self.communication_loop)
            self.receiving_thread.start()
        except:
            print "couldn’t connect to ", server, " on port ", port
            sys.exit()
```
Eye tracker states

• The eye tracker is generally in one of three states:
  - idle, calibrating, running
• Your application (e.g., Unity) controls the tracker
• Two threads run simultaneously:
  - one controls UI and graphics display (UI thread)
  - second accepts streaming data from tracker (ET thread)
• State and gaze data passed through shared memory
  - usually a global object is created
  - tracking thread writes, UI thread reads
Eye tracker communication

- GET and SET XML tags

CLIENT SEND: `<GET ID="TRACKER_DISPLAY" />`
SERVER SEND: `<ACK ID="TRACKER_DISPLAY" STATE="1" />`

CLIENT SEND: `<SET ID="TRACKER_DISPLAY" STATE="1" />`
SERVER SEND: `<ACK ID="TRACKER_DISPLAY" STATE="1" />`

- Read and write to data variables on server
Eye tracking thread

- ET thread sends/receives packets to/from eye tracker
- Gazepoint tracker (Control program) accepts XML data
  - can get/set tracker settings:
    
    ```xml
    CLIENT SEND: <GET ID="TRACKER_DISPLAY" />
    SERVER SEND: <ACK ID="TRACKER_DISPLAY" STATE="1" /> 
    ```
  
  - various other info:
    - TIME_TICK_FREQUENCY
    - SCREEN_SIZE, CAMERA_SIZE
    - PRODUCT_ID, SERIAL_ID
    - etc.
Eye tracking thread

• Application needs to tell tracker what to send
  - ENABLE_SEND_COUNTER
  - ENABLE_SEND_POG_LEFT
  - ENABLE_SEND_POG_RIGHT
  - ENABLE_SEND_POG_BEST
  - ENABLE_SEND_PUPIL_LEFT
  - ENABLE_SEND_PUPIL_RIGHT

• And then tell tracker to stream data

CLIENT SEND: <SET ID="ENABLE_SEND_DATA" STATE="1" />
SERVER SEND: <ACK ID="ENABLE_SEND_DATA" STATE="1" />
SERVER SEND: <REC ... />
Eye tracking thread

• Don’t forget to gracefully shutdown the tracker

CLIENT SEND: <SET ID="ENABLE_SEND_DATA" STATE="0" />

• Processing data:
  - parse XML strings
  - for Gazepoint, mind the ending tokens \r\n
  - these characters must be appended to each string as well

• Trickiest part: calibration
  - need to either call tracker’s calibration routine or
  - synchronize drawing your own calibration points (preferred method)
Calibration

• Eye tracker’s calibration screen (when finished)
Calibration (canned)

• If working on same display as eye tracker control:
  - can use tracker’s own calibration, if available
  - this will take over the screen

• Advantages:
  - eye tracker does synchronization, reports calibration error

• Disadvantages:
  - your app gives up control

```xml
CLIENT SEND: <SET ID="CALIBRATE_SHOW" STATE="1" />
SERVER SEND: <ACK ID="CALIBRATE_SHOW" STATE="1" />

CLIENT SEND: <SET ID="CALIBRATE_START" STATE="1" />
SERVER SEND: <ACK ID="CALIBRATE_START" STATE="1" />
```
Calibration (custom)

• Timing (important!): must synchronize with tracker
  • Two components:
    - time for stationary dot display (seconds)
      · when eye tracking camera samples eye image

```
CLIENT SEND: <SET ID="CALIBRATE_TIMEOUT" VALUE="2" />
SERVER SEND: <ACK ID="CALIBRATE_TIMEOUT" VALUE="2" />
```

- time between dots
  · use this to animate dot moving from point to point

```
CLIENT SEND: <SET ID="CALIBRATE_DELAY" VALUE="1.0" />
SERVER SEND: <ACK ID="CALIBRATE_DELAY" VALUE="1.0" />
```

- start calibration

```
CLIENT SEND: <SET ID="CALIBRATE_START" VALUE="1" />
CLIENT SEND: <ACK ID="CALIBRATE_START" VALUE="1" />
```
Drawing calibration dots (client side)

• If moving, interpolate position of dot
• Can use simple linear function or something more fancy

\[ 60\left( \frac{1}{10}t^5 - \frac{1}{4}t^4 + \frac{1}{6}t^3 \right) \]

Saccade acceleration, velocity, and position profiles

Duchowski, et al. (ETRA 2016)
Drawing calibration dots (client side)

- If moving, interpolate position of dot
- Can use simple linear function or something more fancy

\[60\left(\frac{1}{10}t^5 - \frac{1}{4}t^4 + \frac{1}{6}t^3\right)\]

```python
if calibration.movement == 0:
    t = calibration.timer_elapsed_ms()/calibration.get_moveTime()
    i = calibration.get_dot_index()
    if i+1 < calibration.get_dot_len():
        v1 = calibration.get_dot(i)
        v2 = calibration.get_dot(i+1)
        t = 60.0*((1.0/10.0)*t**5 - (1.0/4.0)*t**4 + (1.0/6.0)*t**3)
        v = ((1.0 - t)*v1[0] + t*v2[0], (1.0 - t)*v1[1] + t*v2[1])
        calibration.set_xy(v)
    if calibration.timer_elapsed_ms() > calibration.get_moveTime():
        calibration.inc_dot_index()
        calibration.timer_start()
        # advance to dilation
        calibration.movement += 1
```
Calibration setup

• Reset calibration to default points

 CLIENT SEND: <SET ID="CALIBRATE_RESET" />
 SERVER SEND: <ACK ID="CALIBRATE_RESET" PTS="5" />

• Or clear out calibration points and add your own

 CLIENT SEND: <SET ID="CALIBRATE_CLEAR" />
 SERVER SEND: <ACK ID="CALIBRATE_CLEAR" PTS="0" />

• Tell eye tracker where you will draw the dots
  - do this for as many points as you want (can do one at a time)

 CLIENT SEND: <GET ID="CALIBRATE_ADDPOINT" />
 SERVER SEND: <ACK ID="CALIBRATE_ADDPOINT" PTS="5" X1="0.50000" Y1="0.50000" X2="0.85000" Y2="0.15000" X3="0.85000" Y3="0.85000" X4="0.15000" Y4="0.85000" X5="0.15000" Y5="0.15000" />
After calibration

• If doing it yourself, can still get calibration report

```xml
CLIENT SEND: <GET ID="CALIBRATE_RESULT_SUMMARY" />  
SERVER SEND: <ACK ID="CALIBRATE_RESULT_SUMMARY" AVE_ERROR="19.43"  
VALID_POINTS="5" />  
```

• Can get more info as calibration proceeds:

```xml
...  
SERVER SEND: <CAL ID="CALIB_RESULT_PT" PT="5" CALX="0.1500"  
CALY="0.1500" />  
SERVER SEND: <CAL ID="CALIB_RESULT" CALX1="0.50000" CALY1="0.50000"  
LX1="0.50229" LY1="0.50279" LV1="1" RX1="0.51467" RY1="0.50870"  
RV1="1" CALX2="0.85000" CALY2="0.15000" LX2="0.84943" LY2="0.14930"  
LV2="1" RX2="0.84600" RY2="0.14763" RV2="1" CALX3="0.85000"  
CALY3="0.85000" LX3="0.84942" LY3="0.84929" LV3="1" RX3="0.84627"  
RY3="0.84779" RV3="1" CALX4="0.15000" CALY4="0.85000" LX4="0.14616"  
LY4="0.84930" LV4="1" RX4="0.84772" RV4="1"  
CALX5="0.15000" CALY5="0.15000" LX5="0.14944" LY5="0.14931" LV5="1"  
RX5="0.14689" RY5="0.14815" RV5="1" />  
```
Calibration result

- Can use the results to reproduce summary screen
Do own calibration

- Use another calibration image (1600 x 900)
Analytics pipeline

• Data processing:

  1. raw data
Analytics pipeline

- Data processing:
  1. raw data
  2. smoothed data

\[
\dot{x}_n^s(t) = \frac{1}{(\Delta t^s)} \left( \sum_{i=-p}^{p} h_i^{t,s} x_{n-i} - \sum_{i=-q}^{q} g_i^{t,s} x_{n-i} \right)
\]
Analytics pipeline

- Data processing:
  1. raw data
  2. smoothed data
  3. fixations (after thresholding)
Analytics pipeline

- Data processing:
  1. raw data
  2. smoothed data
  3. fixations (after thresholding)
Consider accuracy & refitting

- Compute own accuracy and precision estimates
  - include own calibration in stimulus set

- Data allows fit correction via least-squares
Re-calibration plot

- Calibration point
- Original coordinates
- Scaled/shifted coordinates
- Closest centroid

View distance: 25.59 inches, screen: 13 inches, 28.50° (visual angle), dpi: 196.92

Mean error (accuracy): 1.21° (degrees visual angle), standard deviation (precision): 0.53°
Method of least squares

- With \( \{x_{i1}, x_{i2}\} \) and \( \{s_{i1}, s_{i2}\} \) denoting observed and calibration points, resp., set up system of equations:

\[
\begin{bmatrix}
  s_{1x} & s_{1y} \\
  s_{2x} & s_{2y} \\
  \vdots & \vdots \\
  s_{nx} & s_{ny}
\end{bmatrix}
\begin{bmatrix}
  a_0 & b_0 \\
  a_1 & b_1 \\
  a_2 & b_2 \\
  a_3 & b_3 \\
  a_4 & b_4 \\
  a_5 & b_5
\end{bmatrix}
= 
\begin{bmatrix}
  1 & x_1 & y_1 & xy_1 & x_1^2 & y_1^2 \\
  1 & x_2 & y_2 & xy_2 & x_2^2 & y_2^2 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  1 & x_n & y_n & xy_n & x_n^2 & y_n^2
\end{bmatrix}
\]

using \( n \) calibration points
Method of least squares

- Solution is the 2nd order polynomial at each point:

\[
\begin{align*}
    s_x &= a_0 + a_1 x + a_2 y + a_3 x y + a_4 x^2 + a_5 y^2 \\
    s_y &= b_0 + b_1 x + b_2 y + b_3 x y + b_4 x^2 + b_5 y^2
\end{align*}
\]

or in matrix notation: \( S = X\hat{B} \)

following Morimoto and Mimica (2005)

- Generalizes to Lagrange’s method of least square
  (see Lancaster and Salkauskas (1986) for example)
Method of least squares

• Given \( S = X\hat{B} \)

\[
\text{solute for } \hat{B}: \quad \hat{B} = (X^TX)^{-1}X^TY
\]

where \( G^{-1} = (X^TX)^{-1} \) is called the pseudo-inverse

• We may have many fixations at each calibration point
  – use centroid (mean) at each calibration point
  – how to determine which fixations are closest?
  – use \( kd \)-tree
Accuracy / precision

Algorithm:

- for each of $N$ fixation points $P_{i,j}$:
  - using $kd$-tree, find closest of $M$ calibration points $T_i$
  - compute distance to calibration point $\|T_i - P_{i,j}\|$
Accuracy / precision

- Algorithm:
  - compute mean of means (accuracy)
  - compute standard deviation (precision)
  - report error in deg. visual angle (Blignaut and Beelders, 2012)
\( \mathcal{K} \) coefficient: ambient/focal coefficient

\[
\mathcal{K}_i = \frac{d_i - \mu_d}{\sigma_d} - \frac{a_{i+1} - \mu_a}{\sigma_a}
\]

- **Algorithm:**
  - standardized fixation duration - standardized saccade amplitude
  - \( \mathcal{K} > 0 \) suggests focal viewing
  - \( \mathcal{K} < 0 \) suggests ambient viewing (visual search)
Coefficient as contextual cue

- $\kappa$ coefficient is examined over time

- Cognitive load could be indicated by:
  - becoming more focal over time, or
  - oscillating between focal and ambient

- Visual search could be indicated by:
  - largely ambient $\kappa$
Interactive applications

• If you want to roll your own...

- display thread
  - calibrating?
    - Y: sampling?
      - Y: draw dot
        - continue
      - N: move dot
        - continue
    - N: recv coord
      - continue
- eye tracking thread
  - calibrating?
    - Y: send coord
      - continue
    - N: blocking queue

- network
  - XML
  - eye tracking server
Example: designing for gaze interaction

- Using gaze for input: how best to exploit?
  - Midas touch problem known since Jacob (1990)
    - dwell time was established as method for gaze selection
  - Zhai et al.’s (1999) MAGIC pointing’s key observation:
    - the eye is a perceptual organ, not meant for motor input (like hand)
- What about gestures?
Example: designing for gaze interaction

- Example application: gaze-based PIN entry
- Problems with traditional approach:
  - dwell time selection of PIN numerals is slow
  - improve through thoughtful redesign
Background: Gaze-based authentication

- Background:
  - Hoanca and Mock’s (2006) PassFaces idea
    - immediate clear how password would work (intuitive)
    - based on dwell-time, however
Background: Gaze-based authentication

- Background:
  - De Luca et al. (2009) suggest gaze-based graphical password
  - idea is for user to remember shape
  - seems somewhat difficult...
Eye-typing

• Background:
  - would rather use eyes as in *Dasher’s* eye-typing interface
  - reviewed by Tuisku et al. (2008)
  - uses boundary-crossing interface
Gaze-based menus

• Background:
  - following Huckauf and Urbina’s (2008) pEYE menus

- but add central fixation area to exploit center-bias
Modeling interaction

- Fitts’ Law modeling:

\[
\sum_{i} |\text{PIN}| \quad \text{with} \quad \text{MT}_i = a + b \log_2 \left( \frac{2A}{W} \right)
\]
Modeling interaction

• Rotary interface:
  - equidistant targets (numeral *pads*)
  - no need to detect fixations (uses boundary crossings)
  - center-bias exploited with central dot target
Implementation: rotary

- Smoothed boundary crossings yield character stream

- needs to be further processed to give PIN entry code

- use a weighted voted scheme

- emphasize exiting pad

- side-effect:
  - allows correction of initial mass
Implementation: keypad

- Fixation-based implementation straightforward, but...
  - needs start and end fixation ‘tokens’ (* and #)
  - dwell time is required:
    - necessary so that intermediate pads are not selected (Midas Touch)
  - fixation detection is required
  - use Savitzky-Golay filter
  - threshold on velocity
    - “saccade-picker”
Evaluation: user study

- Participants:
  - 30 (24 M, 6 F)
  - ages 18-44
  - data from 3 excluded

- Apparatus:
  - Gazepoint GP3 eye tracker
    - 60 Hz, 1 deg accuracy (according to manufacturer)
  - chin-rest, sturdy (wooden) chair, Dell 19 inch monitor, laptop

- Procedure:
  - 9-point calibration followed by 30-second training
  - each interface tested with 30 trials (order counterbalanced)
Results: speed, accuracy, trials

- Rotary interface: speed, accuracy, trials
  - faster input than keypad interface \( (F(1,26) = 61.20, p < 0.01) \)
  - average PIN error not sig. different \( (F(1,26) = 1.16, p = 0.29, \text{n.s}) \)
  - learning effect sig. only for keypad \( (F(29,754) = 1.96, p < 0.01) \)
Results: speed, accuracy, trials

- Rotary interface: speed
  - faster input than keypad interface
    \( F(1,26) = 61.20, p < 0.01 \)
  - as expected, following Fitts’ Law
  - dwell time clearly a hinderance
  - no need for fixation detection
  - INSTEAD…
  - streaming smoothed gaze data
Results: speed, accuracy, trials

• Rotary interface: accuracy

  • average PIN error not sig. different
    \( F(1,26) = 1.16, p = 0.29, \text{n.s.} \)

• error computed as normalized Levenshtein distance (per PIN)

• usually just one digit error

• BUT...

• overall error rate lower
  \( (28.84\% \text{ vs. } 35.80\% \text{ for the keypad}) \)
Results: speed, accuracy, trials

• Rotary interface: trials
  • learning effect sig. only for keypad
    \( F(29,754) = 1.96, p < 0.01 \)
  • users found rotary UI intuitive
  • users preferred rotary interface

• Some user comments
  - The rotary is quicker and slightly more intuitive.
  - The rotary felt faster to use because I did not have to dwell on the numbers like the keypad.
  - The keypad felt more natural—it was easier and faster.
  - Having to go back to the circle on the rotary seemed like it made me go slower but it’s more efficient in case you mess up—you can change the number before you enter it.
  - The rotary was easier to navigate with my eyes to the different numbers.
Results: keypad vs. rotary

• Similar error rates (no sig. difference; rotary’s overall rate lower)
• Users grasped rotary operation immediately (no learning effect)
• Keypad showed steeper learning curve
• Rotary’s speed entry follows Fitts’ Law model, by design

• Rotary is thus good gaze-based PIN entry candidate
Summary

• Start with good experimental design / hypothesis
• Re-instrument app to “talk” to eye tracker
• Calibrate, calibrate, calibrate (still)
• Analysis involves filtering of raw data to detect fixations
• Start with visualization (sanity check)
• Analyze data (find fixations, collate, stats)
• Write up report
• Automate as much as possible
Q & A

• Thank you!

• Questions?
Eye Tracking

Select References
References


References

References


- Peysakhovich, V. Study of pupil diameter and eye movements to enhance flight safety. Ph.D. thesis; Université de Toulouse; Toulouse, France; 2016.


References


References


References


• Grinding, T. Eye Movement Analysis & Prediction with the Kalman Filter. Master’s thesis; Clemson University; Clemson, SC; 2006.

References


References


References

- Howard, IP, Rogers, BJ. Seeing in Depth; vol. II: Depth Perception. Thornhill, ON, Canada: I Porteous, University of Toronto Press; 2002.
References


References


References


References


References


References


