MODULAR STREAMING PIPELINE OF EYE/HEAD TRACKING DATA USING TOBI PRO GLASSES 3

Hamed Rahimi Nasrabadi
Department of Biological and Visual Sciences
SUNY College of Optometry, New York, NY 10036
hraiminasrabadi@sunyopt.edu

Jose-Manuel Alonso
Department of Biological and Visual Sciences
SUNY College of Optometry, New York, NY 10036
jalonso@sunyopt.edu

ABSTRACT

Head-mounted tools for eye/head tracking are increasingly used for assessment of visual behavior in navigation, sports, sociology, and neuroeconomics. Here we introduce an open-source python software (TP3Py) for collection and analysis of portable eye/head tracking signals using Tobii Pro Glasses 3. TP3Py’s modular pipeline provides a platform for incorporating user-oriented functionalities and comprehensive data acquisition to accelerate the development in behavioral and tracking research. Tobii Pro Glasses 3 is equipped with embedded cameras viewing the visual scene and the eyes, inertial measurement unit (IMU) sensors, and video-based eye tracker implemented in the accompanying unit. The program establishes a wireless connection to the glasses and, within separate threads, continuously leverages the received data in numerical or string formats accessible for saving, processing, and graphical purposes. Built-in modules for presenting eye, scene, and IMU data to the experimenter have been adapted as well as communicating modules for sending the raw signals to stimulus/task controllers in live fashion. Closed-loop experimental designs are limited due to the 140ms time delay of the system, but this limitation is compensated by the portability of the eye/head tracking. An offline data viewer has been also incorporated to allow more time-consuming computations. Lastly, we demonstrate example recordings involving vestibulo-ocular reflexes, saccadic eye movements, optokinetic responses, or vergence eye movements to highlight the program’s measurement capabilities to address various experimental goals. TP3Py has been tested on Windows with Intel processors, and Ubuntu operating systems with Intel or ARM (Raspberry Pie) architectures.

Introduction

Throughout the course of evolution, most of visual animals developed eye and head motion to efficiently sample their visual environment. Eye motion frequently involves fast transitions in eye rotation known as saccades, which are common in visual animals with or without high-resolution foveas in their retinas [Kowler 2011, Land 1999, Meyer et al. 2020, Wallman and Pettigrew 1985]. In addition, animals with foveal vision can generate smooth pursuit eye movements when fixating and tracking a moving target. In everyday tasks, humans coordinate their head and eye movements while making voluntary saccadic or pursue eye motions to inspect specific regions of the visual field [Bizzi 1974, Lanman et al. 1978]. They also generate compensatory eye movements through vestibulo-ocular reflexes (VOR), which are evolutionary well-preserved and neutralize head movements with eye rotations to keep the retinal image stable and eliminate motion blur [Miles and Lisberger 1981, Raphan and Cohen 2002]. The optokinetic reflex (OKR)
also allows to maintain a stable gaze in a moving environment [Huang and Neuhaus, 2008, Konen et al., 2005].

Head-mounted tracking devices have been used in numerous studies to understand our diverse visual steering strategies within the scopes of sports and navigation [Bahill and LaRitz, 1984, Fegatelli et al., Mann et al., 2007, Shater et al., 2011], social interactions [Fiedler et al., 2013, Jarodzka et al., 2021, Macdonald and Tatler, 2018, Pfeiffer et al., 2013, Wieser et al., 2009], neuroeconomics [Krucien et al., 2017, Polonio et al., 2015, Reutskaja et al., 2011, Van Der Lans et al., 2008], and cognitive and visual impairments diagnosis [Harezlak and Kasprowski, 2018, Tao et al., 2020, Vidal et al., 2012, Kullmann et al., 2021]. Accommodation, pupil dilation, and convergence further expand the diversity of our visual behavior under different demands of attention, decision making, and depth perception [Fincham and Walton, 1957, Parker, 2007, Preuschoff et al., 2011, Sole Puig et al., 2013, Hoffman et al., 2008, Gibaldi and Banks, 2019].

Within past few decades, software tools for eye tracking advanced substantially with the use of video-based algorithms for gaze estimation [Rufa et al., 2005, Yu and Eizenman, 2004, Cognolato et al., 2018, Lee et al., 2012, Wade and Tatler, 2005, Kim et al., 2004]. This progress occurred in parallel with technological improvements in portable devices and head-mounted displays (HMD), which led to development of more efficient eye and head tracking methods. We chose Tobii Pro Glasses 3 (released in 2020) to perform portable eye tracking, scene capturing, and pupillometry that satisfies our precision criteria for studying naturally-moving visual behavior. Here we introduce TP3Py, an open-source python controller program for live collection and analysis of Tobii Pro Glasses 3 data. Our goal for building the library was to provide a flexible platform for accelerating research in eye/head tracking and behavioral studies. TP3Py allows users to employ custom processing/graphical modules using python programming language. It also allows saving comprehensive behavioral data solely in the host computers including eye movies which are not currently saved within Tobii’s API. The paper is organized into four sections. In section 1, we describe the specifications of Tobii Pro Glasses 3. In section 2, we describe the implementation of TP3Py and built-in modules. In section 3, we explain the calibration procedures. In section 4, we provide a few example recordings and discuss the research directions where the program could have an impact.

1 Tobii Pro Glasses 3 Description

Tobii Pro Glasses 3 have similar weight to conventional glasses (77 grams), but are equipped with multiple sensors and cameras to perform eye tracking while freely moving the head (table 1). A HDMI cable transmits the sensor’s output from the glasses to the accompanying recording unit, where video-based eye tracking is performed. The system infers the eye’s direction with 0.6 degrees accuracy by processing the corneal reflection produced by infrared illuminators placed on the glasses (8 per eye). Tobii outputs eye tracking data in a gaze data structure that includes gaze position (gaze2D) in the scene video frame, 3D gaze location for both eyes, and pupil diameters and gaze directions of right or left eyes (table 2). The scene camera’s field of view is 95 and 63 degrees in horizontal and vertical directions. Gaze2D variable outputs (0, 0) or (1, 1) if the subject is looking at the top left corner or bottom right corner of the scene video frame. Tobii is accompanied with rechargeable batteries, each with recording time of 105 minutes.

Tobii Pro Glasses 3 transmit multiple digital media packages over wireless via the Real Time Streaming Protocol (RTSP) to the applications. The data packages are specified by the Real-time Transport Protocol (RTP) where distinct data modalities are assigned to a specific payload identifier. The payload identifiers carry information regarding the encoding format, sampling size and sampling frequency of the transmitted data packages. Tobii arranges its data outputs within 6 dynamic payloads from 96 to 101. For instance, scene video frames and scene audio are transmitted over payload 96 and 97. Gaze data structure is transmitted over payload 99 with 100Hz or 50Hz sampling frequency and IMU data is streamed over payload 100. Tobii’s recording unit is also accompanied with a Transistor-Transistor Logic
Table 1: Embedded sensors of Tobii Pro Glasses 3

| Sensors        | Sampling frequency | Details                                           | RTP Payload | Encoding format |
|----------------|--------------------|---------------------------------------------------|-------------|-----------------|
| Scene Camera   | 25Hz               | 1920x1080 pixels, YUV420p format 96 and 97        | 96 and 97   | H.264           |
| Eye Cameras    | 50Hz or 100Hz      | Two cameras per eye each 256x256 pixels, Single channel | 98          | H.264           |
| Accelerometer  | 100Hz              | 3 axes                                           | 100         | ASCII           |
| Gyroscope      | 100Hz              | 3 axes                                           | 100         | ASCII           |
| Magnetometer   | 10Hz               | 3 axes                                           | 100         | ASCII           |

Table 2: Gaze Data structure

| Gaze data            | Variable Keyword | Details                                           |
|----------------------|------------------|---------------------------------------------------|
| Measurements for both eyes | gaze2D          | Gaze location in the scene video frame coordinates |
| 2D Gaze point        |                  |                                                   |
| 3D Gaze point        | gaze3D           | Gaze location in egocentric space in mm           |
| Measurements for each eye | pupildiameter  | Pupil diameter for each eye                       |
| Pupil diameter (mm)  | gazedirection    | A unit 3D vector signaling the gaze direction of each eye |
| 3D gaze direction    |                  | Location of each eye in mm in egocentric space    |
| 3D gaze origin       | gazeorigin       |                                                   |

(TTL) circuit which detects a change in the logical value of the TTL input (high to low or low to high) and sends the events over payload 101 that can be used for synchronizing the recording device to stimulus/task controllers.

2 TP3Py Program Design

Here we developed an open-source python platform for establishing a flexible communication bridge between Tobii Pro Glasses 3 and client applications. Moreover, the program allows saving comprehensive data including eye videos in the computer host without a need for SD memory cards. We used Gstreamer backend libraries (originally written in C++) to establish the connection and handle the received packages from RTSP server. We also incorporated parallel computing and a Graphical User Interface (GUI) in the TP3Py by using PyQt5 library. TP3Py allows users to develop custom-made processing functions or modules to address a variety of experimental designs and processing objectives.

Figure 1a demonstrates the main interface of the program. Prior to running an experiment, Tobii Glasses should be turned on and connected with WiFi (password TobiiGlasses). The ‘Initials’ and ‘Experiment Name’ text fields correspond to the name/initials of the subject and the conditions of experiment that is being tested. The subject needs to look at the calibration target positioned at 50cm to 100cm distance in front of the eyes after the calibration button is pushed. If calibration is successful, the calibration status becomes green. Calibration can also be performed in the middle of the experiment; however, the data transmission will be temporarily paused. The panel on the right side of the main interface indicates what modules are included in the experiment. Modules access the streaming signals provided by the glasses in live fashion and execute custom processing, graphical, and communicating functions that may be handled within separate threads (fig 1b). For instance, communication modules continuously transfer the acquired signals to other computers where the received information is used for controlling the task conditions. The modules cannot be added at the middle of the experiment. At the end of the experiment, a time period of 5 seconds is needed to make the program ready for the next experiment.
Figure 1: **Main Program’s GUI.** a. Main graphical interface of the TP3Py including text fields to enter the participant’s initials and experiment name, start/end buttons, eye calibration button and its status (green if calibrated and red if calibration is needed), and module panel for adding and removing modules in the program’s pipeline. b. A schematic of TP3Py’s modular pipeline where each module can access distinct data types (i.e., eye frames every 2ms) and execute custom functions for graphical, analysis, and communicative purposes.

2.1 Implementing Gstreamer

The main body of Gstreamer contains a pipeline that controls the flow of the application and data pre-processing performed on the received raw packages. We used rtspsrc element at the beginning of the Gstreamer pipeline to connect to the RTSP streaming source. Processing on each RTP packet identified by the payload number are then followed within parallel branches in the pipeline. We used rtph264depay element to extract videos with H.264 format from RTP packets [Wenger et al., 2005] containing the scene and eye video (96 and 98 payloads). Following extraction, we implemented two parallel branches (using tee element) for saving and reading video frames. In the saving passage, the H. 264 formatted videos are directly saved using splitmuxsink element. The splitmuxsink element saves the video in patches of a specified length (default value of a minute) in the .mp4 format. In the reading branch, we employed avdec_h264 element for decoding the video frames and parsed the outcome to the appsink element which makes the received frames accessible to the program in numerical array. Frame’s numerical arrays are then reshaped according to their corresponding image format. For instance, scene images (YUV420p format) are reshaped into a matrix with size of (frame_height*3/2, frame_width). Segmenting the video files in the saving branch makes it easier to load the videos for later usages since the video recorded within the time course of an hour can be immense in size and slow to load. We saved the timestamps of the first video segment frame for time synchronization and aligning the segments. Moreover, all the frame timestamps in the video reading branches are saved.

Appsink element has been also used as an endpoint element of Gstreamer pipeline when processing the rest of the payloads such as IMU data (payload 100), Gaze data (payload 99), or TTL (payload 101) without additional pre-
processing. The advantage of using appsink element is that it leverages the raw data in a form of buffers accompanied by the presentation timestamp (PTS) that can be used to synchronize different signals. Video frame buffers are converted to numerical arrays (numpy arrays) while other sources are converted to strings (JSON format) which are later used for saving, processing, and graphical threads. PTS times are generated by a clock that can be set by the Tobii’s internal clock or a local clock of the application host. We will explain the timing characteristics of the system in section 2.5.

2.2 Threading, Visualization, and User Interface

We incorporated PyQt5 library for implementing a graphical user interface (GUI) in TP3Py’s pipeline. Not only PyQt5 helped with the front-end visualization but also brought multi-threading functionalities which we used to enhance the performance of the program. Each thread or worker in TP3Py is instantiated from the PyQt5’s QThread class. PyQt5’s most prominent feature is the pyqtSignal and pyqtSlot attributes which allow different threads to communicate. For a connected signal/slot pair, the slot function is called when the signal is emitted, performing a defined function on the received signal. Signals may contain only a digit, number array (numpy ndarray), or characters (string) and they are emitted once a new data acquired or a custom criterion has passed. The Gstreamer thread as an example, continuously uses the pyqtSignals to emit the streamed data to the slots of the other threads that need to carry out saving, processing, or visualizing tasks. Thread classes cannot terminate themselves and they have to send a pyqtSignal to the threads that initialized them for quitting and clearing the memory.

All graphical aspects of the TP3Py are handled with PyQt5 library via the main GUI thread. The main thread manages the user’s input, and it initializes the Gstreamer, saver threads, and other custom-made modules. We utilized PyQt5 widgets for creating text fields, push buttons and other user interfaces in the main window and module interfaces. We further implemented pynput library for inputting keyboard events during the experiment.

2.3 Saver Threads

We employed saver threads to continuously save the received data in text files. We connected Gstreamer’s pyqtSignals that contained IMU, Gaze, or timestamps data to the corresponding IMU, Gaze, or time saver-threads. For each experiment all files are saved in a folder named with the experiment’s starting time in the subject name’s directory. All eye video file names are formatted according to ‘eye-ExpName-hour-minute-count’, where ExpName is the input given by the experimenter (figure 1), hour-minute is initial time of the recording, and the count is the number of the video in the segmented sequence (count 0 is the first video). Gazedata, IMU, and video timestamps/TTLs are saved in JSON format in the text file with file names beginning with ‘Gazedata-ExpName-hour-minute.txt’, ‘IMUdata-ExpName-hour-minute.txt’, ‘TSdata-ExpName-hour-minute.txt’. Table 3 presents the variables being saved in each text file.

2.4 Built-in Modules

Custom-made python codes or modules can be incorporated into the TP3Py’s pipeline. The flexibility in accessing Gstreamer signals in addition to the threading capabilities, allow modules to address variety of objectives spanning graphical, analysis, saving, and communicating purposes. Each module is a python class which inherits methods and properties from PyQt5 QOobject class and takes the Gstreamer class as an input. Modules can then connect to the Gstreamer signals for accessing the Gstreamer output data such as scene/eye video frames, IMU, and gaze data (fig 1b). Module to module connections can also be established for communicating the processed data if linked properly. Modules are generally organized by a main component and a graphical component. Main components contain the class definition, threading initialization, and callable functions (slots) for processing the received data from other modules/workers. The graphical components characterize the visual interface and layout of the modules. Solely graphical modules generally are not considered as separate threads, since PyQt5 graphics should be rendered in only
Table 3: JSON Variables of Gazedata, IMUdata, and TSdata files

| Data File     | Variable | Details                                      |
|---------------|----------|----------------------------------------------|
| Gazedata- tgz | eyeright | PTS of received gaze data structure          |
|               | eyeleft  | Data from left eye                           |
|               | Gaze data structure | According to table 2                                    |
| IMUdata- tacc | accelerometer | PTS of received accelerometer and gyroscope data |
|               | gyroscope | Linear acceleration data in x, y, and z axes  |
|               | tmag     | PTS of received magnetometer sensor          |
|               | magnetometer | Magnetic field intensity in x, y, and z axes |
| TSdata-       | eyeMSTS  | PTS of the first frame of eye movie segment |
|               | sceneMSTS| PTS of the first frame of scene movie segment |
|               | ttseye   | Time of the eye video frame                  |
|               | tssscene | Time of the scene video frame                |
|               | tevent   | Time of the keyboard input                   |
|               | ttl      | PTS of the TTL signal                       |
|               | direction| Direction of TTL on whether input or output signal |
|               | value    | Value of the TTL signal                     |

the main application thread (other graphical libraries such as OpenCV may allow threading without directly using PyQt5’s visualization methods). Nevertheless, Module’s processing loads can be executed within dedicated thread, and this is useful when creating communicating and processing modules.

We incorporated graphical modules for presenting eye videos, scene video and IMU data during the experiment (Figure 1). We converted the color space of the scene (YUV420p) and eye (Gray) movie frames to RGB using opencv library for presenting the video frames into the image canvas of the ShowScene and ShowEyes modules. We also incorporated matplotlib library for plotting IMU data in IMUplot module. Furthermore, GazeMessage and TimingTest modules are communicating modules where gaze and TTL data structures are continuously sent to the linked computers over TCP/IP protocol in dedicated threads.

### 2.5 Timing Characteristics

For measuring the timing performance of the program, we employed a closed loop design to measure the time delay between sending and receiving signals. We wrote a Matlab code to send logical (3.3v) triggers using a Data Acquisition Card (NI USB-6001) and 3.5 mm jack adapter connected to Tobii’s recording unit. Upon receiving the triggers, the TP3Py’s TimingTest module sends a character through TCP/IP protocol to the Matlab controller. The controller then finds the time delay between the trigger onset and the time of received character from TP3Py using LAN connection. All the delays in the measurements are averaged across 10 recording sessions each spanning 30 TTL repeats separated within 3 seconds.

The ‘buffer-mode’ variable of the rtspsrc element in the Gstreamer controls the timing conditions. Time reference for data timestamps (PTS) can be synchronized by the sender clock (buffer-mode: 1), internal clock of application host (buffer-mode: 4), or RTP timestamps (buffer-mode: 0). When timestamps are synchronized by the internal application clock, the program is slow (2s). A faster connection is accomplished by synchronizing to the sender clock (0.5 s). However, the timing quality of the data decreases substantially due to interrupted timestamps. The best timing performance is achieved by having the time reference set by RTP timestamps (mean: 140 ms, std: 14ms). RTP timestamps are the timestamps when the data is recorded on the Tobii device, and they differ from the sender clock...
since Tobii may send data packets at different times irrespective of data acquisition time. Except for the first timestamp, all PTS synchronized to RTP timestamps increase monotonically. Using video modules does not affect the time delay of the program. However, plotting modules such as IMUplot increase the time delay to 180ms. The variations in the timings of the TTL timestamps does not propagate in the relative time differences of TTL’s onsets and offsets (below millisecond accuracy). This suggest that part of the variation in the absolute time measures is inherited in the buffer delay of the DAQ boards connected to the controller computers [Hwang et al., 2019].

All experiments including timing tests are conducted while running TP3Py on laptop (Razor blade, Intel core i7 2.3 GHz, 32 GB RAM) with Ubuntu (20.04) operating system. TP3Py’s open-loop performance has been also tested on windows operating system and portable computing systems with ARM processing architecture (Raspberry Pie 4 model B 8GB RAM). Streaming time delay in Raspberry Pie computing devices are similar to the laptops (150 ms). However, due to the limit in the processing power of the Raspberry, one would need to reduce the scene and eye video frame sizes only in the reading branch of the Gstreamer pipeline without affecting the quality of saved videos.

3 Sensor Calibration

Tobii glasses are equipped with inertial and camera sensors that need to be calibrated prior to data analysis. Moreover, the Tobii’s embedded eye tracker requires a single-point calibration procedure for accurate computation of gaze data. Here we provide details on the calibration protocols we performed on the sensors.

3.1 InertialSensors

Figure 2a shows the spatial coordinates in which the inertial sensors and eye tracking are based upon. We first calculated the bias and standard deviation of the gyroscope, accelerometer, and magnetometer measurements for each x, y, and z axis while keeping the glasses in a fixed location. The bias and std measurements are later used for estimating head angle using various algorithms. Subtracting average bias from gyroscope data did not result in a Gaussian distribution of the measurements due to the drift present in the gyroscope sensors (figure 2b).

Magnetometers face substantial distortions in their output due to magnetic effects of the surrounding electrical circuit. Permanent magnetic fields in the circuit (hard iron) adds a constant bias to the measurements while other ferromagnetic materials may modulate the magnetic field in particular directions (soft iron). For measuring the hard and soft iron effects on the magnetometer, we slowly rotated the glasses within 8-figure shaped trajectories in various directions. Given the symmetry in the above trajectories, all measurement axis should maintain similar statistics (mean and std). However, due to the hard/soft iron effects, some axes have larger intensity on average than others (figure 2c). In order to counteract these biases, we applied a linear transformation based on the percentile values on each measurement axis to transform the elliptical scatter plots to symmetrical spherical shape centered at zero (figure 2d).

3.2 Scene Camera

Spatial relationships between several points in the scene are not preserved in the captured video frames due to the distortions produced by the scene camera optics and sensors. For instance, the length of a bar oriented horizontally or vertically may differ in the captured frames. An intrinsic camera matrix addresses the shift and scale differences in horizontal and vertical directions. Moreover, radial and tangential distortions are produced by the lenses and their displacement relative to the camera sensor. For finding the intrinsic camera matrix and distortion coefficients, one has to calibrate the camera only once.
Figure 2: **Calibration procedures** a. A graph illustrating the spatial coordinates of the glasses. Circular arrows represent gyroscope measurements of rotation speed around the corresponding axes. b. Histogram of the unbiased gyroscope measurements along X, Y, and Z axis. c. Scatter plot of the raw magnetic field intensity measured along XY, YZ, and XZ axis pairs. d. Resulted data scatters after removing hard/soft iron biases from magnetometer sensor. e. Schematic of the procedure used for calibrating scene camera by taking snapshots of the checkerboard pattern viewed at different distances and orientations. f. Using calibrated scene images to infer the head’s relative position and viewing angle. Solid white circles represent the display corners used for estimating the heading direction. The display coordinates are represented by arrows (the thick arrow represents the Z axis). g. Calibration target used in Tobii’s eye tracker. Target background should be white. Inner, middle and outer diameters are 3mm, 23mm, and 43mm correspondingly.

For camera calibration, we generated 8x8 checkerboard stimulus patterns and captured the scene video of the stimulus from various positions and viewing angles, while slowly moving the camera (figure 2e). Checker’s width and height were set equally to 100 pixels in the display. By knowing the locations of the checker’s intersection points in the display’s coordinates and their corresponding locations in the captured frames, we estimated the calibration coefficients of the scene camera using calibrateCamera function of OpenCV library [Zhang, 2000]. Calibrating the scene camera helps preserve the spatial properties of the scenes which could be used for estimating camera’s location and orientation (figure 2f). In display-based task where the width and height of the monitor is known (the relative locations of the corner points), one could estimate the location and head angle of the observer relative to the display by using the calibrated scene video [Marchand et al., 2016]. Moreover, scene videos provide optic flow measurements that can be also used to infer the head’s direction and its association with the perception [Burlingham and Heeger, 2020].

### 3.3 Eye Tracker Calibration

Tobii’s embedded eye tracker performs a single-target calibration procedure. Upon requesting eye calibration, Tobii detects a unique target (figure 2g) in the scene image and corrects for the eye position biases and drifts. The nominal accuracy of the eye tracking reaches 0.6 degrees. For initializing the calibration, we established a connection to the Tobii API through WebSocket protocol in TP3Py. The calibration target is placed within 50cm to 100cm in front of the observer for 2-4 seconds. The Tobii API then sends a JSON string signaling whether the calibration was successful or not.
4 Data Viewers

We developed data viewers in Python and Matlab for viewing and performing post-hoc data analysis. Python data viewer allows previewing the scene and eye videos in a synchronized fashion using the saved timestamps of the scene and eye video frames. We also incorporated image processing functions on the eye images for finding the pupil center and fitting ellipsoids to the pupil shapes that can be utilized for implementing a simple pupil-based eye tracker. Moreover, all data are accompanied by dedicated timestamps which helps to synchronize distinct measurements with high timing accuracy (<1 ms).

4.1 Extracting Head/Vergence Angles

We employed sensor fusion algorithms for estimating the head angle using accelerometer, gyroscope, and magnetometer sensors [Guo et al., 2017, Madgwick et al.]. Since the magnetometer sensor is sampled at 10 Hz, we interpolated the measurements to 100Hz to align with inertial sensors. Prior to fusion, all sensors are calibrated using the procedures explained in 3.1. We further measured the vergence angles as follows. Given a unit gaze vector ($\vec{g}$) of left or right eye, we calculated the horizontal ($\theta$) and vertical ($\phi$) eye angles using the following equation:

$$\theta = \arctg\left(\frac{g_x}{g_z}\right), \phi = \arctg\left(\frac{g_y}{g_z}\right)$$

We then calculated the convergence angle of two eyes in horizontal ($\Theta$) and vertical ($\Phi$) planes by subtracting eye angles above.

$$\Theta = |\theta_{left} - \theta_{right}|, \Phi = |\phi_{left} - \phi_{right}|$$

One could also measure the 3D vergence angle ($\Delta$) of the eyes by finding the angle between left and right gaze directions using dot product as below (gaze directions are unit vectors). Locations of the eyes (in the egocentric coordination) are also provided by Tobii which could be used for measuring the viewing distance. However, one could use the z component of the Gaze3D variable which represents the gaze depth in mm.

$$\Delta = \arccos(\vec{g}_{left}.\vec{g}_{right})$$

4.2 Sample Recordings

We show an unfiltered sample of a recording session in figure 3. In this experiment, the observer fixated on a target placed at 60 cm and rotated his head sequentially in right/left, up/down, and clockwise/counterclockwise directions. Head movements produced counteracting eye movements to minimize the retinal image drift and avoid motion blur. The elicited horizontal VOR eye movements are correlated with the head rotation ($r$: 0.62, p-value « 0.005) with time delay of 5 ms (using cross-correlation) consistent with other measurements [Ramachandran and Lisberger, 2005]. Furthermore, changes in the head’s pitch angle alters the pupil diameter as well as the horizontal vergence angle signaling an association between VOR and vergence oculomotor afferent signals [Cova and Galiana, 1996].

We measured timing regularities of the recorded data by accumulating time durations of missed data packages and normalizing by the total experiment time (missed rate). Scene and eye videos showed the highest timing consistency while IMU data showed the lowest. Abrupt head movements and closing the eyes are the main contributors to loosing eye tracking data. It is recommended to adjust the glasses using a proper nose pad for centering the eyes on the eye video frames for better tracking consistency, especially when there are large eye movements.

We further conducted three more recordings, the results of which are presented in the figure 4. In the first recording, the observer had to maintain fixation on a target moving in depth from 10 cm to 100 cm from the observer. The
Figure 3: Example VOR recording. a. 2D gaze location of the observer in the scene video plotted across time. b. Horizontal gaze angles of left and right eyes over time. c. Pupil diameter of left and right eye in mm over time. d. Convergence angle of two eyes in horizontal and vertical planes over time. e. Head’s pitch, roll, and yaw angles in degrees over time using Fast Kalman algorithm [Guo et al., 2017]. f. Presentation Time Stamp differences of distinct streaming media including gaze data (gaze), IMU, scene video (SV), and eye video (EV). Scene and eye videos have the highest timing regularities (0% missed rate) followed by the gaze data with 2% missed rate while IMU data showed the highest timing irregularity (6% missed rate).

measurements of convergence angle between the two eyes closely follow the convergence eye movement associated with the changes in target location over time. In the next recording (figure 4b), we derive optokinetic responses induced when an observer passively looks at a large (10 deg. size) moving grating stimulus (90 deg. orientation). The pattern of smooth eye movements followed by sudden resetting saccades reproduce the optokinetic nystagmus. In the last recording (figure 4c), we presented two square targets separated by 3 visual degrees horizontally. The subject elicited saccadic eye movements that alternated the gaze location in two targets. Saccadic movements are mediated naturally without imposing any restrictions. Therefore, the results highlights the importance of fast gaze transitions not only when viewing natural scenes but also when assessing simple visual environments [Fabius et al., 2016, Samonds et al., 2018].

Future Directions and Discussion

We developed a modular controller for acquisition and analysis of multi-modal data including eye/head motion, pupil diameter, convergence, and visual scene content, which helps to provide comprehensive and quantitative measures of
Figure 4: Example recordings of oculomotor vergence, nystagmus, and saccadic motion. a. Horizontal and vertical convergence angle of two eyes over time when an observer fixates on a moving target in depth. b. Horizontal gaze angles of left and right eyes over time when observer views a large moving drum stimulus inducing horizontal optokinetic nystagmus. c. Horizontal gaze angles of left and right eyes over time when subject is presented with two square targets in the visual field. Green and black vertical dashed lines represent the stimulus onset and offset indicated by switching the TTL’s state. The experiment took 20 minutes to complete, however, time window of 15 seconds has been shown.

visual behavior without imposing restrictions on the head movements. Incorporating TP3Py in open-loop experimental designs provides helpful information on visual behavioral studies. Therefore, the program may help extend the scope of analysis and potential findings when studying psychology, visual perception, and human behavior [Rahimi-Nasrabadi et al., 2021]. The controller’s flexibility and a short time delay (140 ms which is less than a blink duration), allows the program to accommodate various closed-loop experimental designs. TP3Py may be then used for visual rehabilitation and visual training where appropriate feedbacks are required [Trauzettel-Klosinski, 2011, Hernández-Rodríguez et al., 2020] or in studies where visual behavior are mediated in a controlled fashion.

The advancements in portable eye/head tracking devices impact the landscape of freely moving behavioral research. New investigations are needed to fill our current shortfalls on how distinct types of eye/head motions interact in natural setting and what are their consequences for the visual perception [Bouvier et al., 2020, Nau et al., 2018, Shiozaki and Kazama, 2017, Vélez-Fort et al., 2018]. Dynamics of the high-dimensional behavioral motion steered by the natural viewing experiences are more closely addressed when using portable devices [Einhauser et al., 2007, Lappi, 2016, Hayhoe and Ballard, 2005]. As we demonstrated in the example recordings using TP3Py, one can probe the head motion and visual world’s effects on the eye movements, allowing for building more naturalistic image of visual behavior.
TP3Py may also help to accelerate the research in creating novel human-computer interfaces and machine learning tools for extracting relevant information from the visual behavior [Klaib et al., 2021, Kollias et al., 2021, Zemblys et al., 2018, Mildenhall et al., Lee et al., Srivastava et al., 2022]. TP3Py is developed in python programming language which is widely used by the machine learning community and has proved itself efficient for performing artificial intelligence algorithms. Lastly, the software may assist the research in developing video-based eye/head tracking techniques and it may provide benchmarking tools for developing artificial agents that are not only influenced by verbal communications, but also elicit visual and behavioral interactions [Gan et al., Srivastava et al., 2022].

Acknowledgments

The study was supported by NIH grant EY027361. We thank Vandad Davoodnia for helping to develop the module importation UI.

Code Availability

The source code of the TP3Py controller and data viewers can be accessed in the link below:

Link: https://github.com/SUNYOpt/TP3Py

Conflict of Interest and Ethics Approval

No potential conflict of interest was reported by the authors. All experiments in human subjects were approved by the institutional review board at the State University of New York College of Optometry and followed the principles outlined in the Declaration of Helsinki. Consent from all participants was obtained prior to the experiments.

References

E. Kowler. Eye movements: the past 25 years. Vision Res, 51(13):1457–83, 2011. ISSN 1878-5646 (Electronic) 0042-6989 (Linking). doi:10.1016/j.visres.2010.12.014 URL https://www.ncbi.nlm.nih.gov/pubmed/21237189.

M. F. Land. Motion and vision: why animals move their eyes. Journal of Comparative Physiology A, 185(4):341–352, 1999. ISSN 1432-1351. doi:10.1007/s003590050393 URL https://doi.org/10.1007/s003590050393.

Arne F. Meyer, John O’Keefe, and Jasper Poort. Two distinct types of eye-head coupling in freely moving mice. Current Biology, 30(11):2116–2130.e6, 2020. ISSN 0960-9822. doi:https://doi.org/10.1016/j.cub.2020.04.042 URL https://www.sciencedirect.com/science/article/pii/S096098222030556X.

J. Wallman and J. D. Pettigrew. Conjugate and disjunctive saccades in two avian species with contrasting oculomotor strategies. J Neurosci, 5(6):1418–28, 1985. ISSN 0270-6474 (Print) 0270-6474 (Linking). URL https://www.ncbi.nlm.nih.gov/pubmed/4009240.

E. Bizzi. The coordination of eye-head movements. Sci Am, 231(4):100–6, 1974. ISSN 0036-8733 (Print) 0036-8733 (Linking). URL https://www.ncbi.nlm.nih.gov/pubmed/4216072.

J. Lanman, E. Bizzi, and J. Allum. The coordination of eye and head movement during smooth pursuit. Brain Res, 153(1):39–53, 1978. ISSN 0006-8993 (Print) 0006-8993 (Linking). doi:10.1016/0006-8993(78)91127-7 URL https://www.ncbi.nlm.nih.gov/pubmed/98220.

F. A. Miles and S. G. Lisberger. Plasticity in the vestibulo-ocular reflex: A new hypothesis. Annual Review of Neuroscience, 4(1):273–299, 1981. ISSN 0147-006X. doi:10.1146/annurev.ne.04.030181.001421 URL https://doi.org/10.1146/annurev.ne.04.030181.001421.
Theodore Raphan and Bernard Cohen. The vestibulo-ocular reflex in three dimensions. *Experimental Brain Research*, 145(1):1–27, 2002. ISSN 1432-1106. doi:10.1007/s00221-002-1067-z URL https://doi.org/10.1007/s00221-002-1067-z

Y. Y. Huang and S. C. Neuhauss. The optokinetic response in zebrafish and its applications. *Front Biosci.*, 13:1899–916, 2008. ISSN 1093-9946 (Print) 1093-4715 (Linking). doi:10.2741/2810 URL https://www.ncbi.nlm.nih.gov/pubmed/17981678.

Christina S. Konen, Raimund Kleiser, Rüdiger J. Seitz, and Frank Bremmer. An fmri study of optokinetic nystagmus and smooth-pursuit eye movements in humans. *Experimental Brain Research*, 165(2):203–216, 2005. ISSN 1432-1106. doi:10.1007/s00221-005-2289-7 URL https://doi.org/10.1007/s00221-005-2289-7.

A. Terry Bahill and Tom LaRitz. Why can’t batters keep their eyes on the ball? *American Scientist*, 72(3):249–253, 1984. ISSN 0003-0996 (Print).

Dario Fegatelli, Francesco Giancamilli, Luca Mallia, Andrea Chirico, and Fabio Lucidi. The use of eye tracking (et) in targeting sports: A review of the studies on quiet eye (qe). In Giuseppe De Pietro, Luigi Gallo, Robert J. Howlett, and Lakhmi C. Jain, editors, *Intelligent Interactive Multimedia Systems and Services 2016*, pages 715–730. Springer International Publishing. ISBN 978-3-319-39345-2.

D. T. Mann, A. M. Williams, P. Ward, and C. M. Janelle. Perceptual-cognitive expertise in sport: a meta-analysis. *J Sport Exerc Psychol*, 29(4):457–78, 2007. ISSN 0895-2779 (Print) 0895-2779 (Linking). doi:10.1123/jsep.29.4.457 URL https://www.ncbi.nlm.nih.gov/pubmed/17968048.

Daniel M. Shafer, Corey P. Carbonara, and Lucy Popova. Spatial presence and perceived reality as predictors of motion-based video game enjoyment. *Presence: Teleoperators and Virtual Environments*, 20(6):591–619, 2011. doi:10.1162/PRES_a_00084 URL https://doi.org/10.1162/PRES_a_00084.

Susann Fiedler, Andreas Glöckner, Andreas Nicklisch, and Stephan Dickert. Social value orientation and information search in social dilemmas: An eye-tracking analysis. *Organizational Behavior and Human Decision Processes*, 120(2):272–284, 2013. ISSN 0749-5978. doi:https://doi.org/10.1016/j.obhdp.2012.07.002 URL https://www.sciencedirect.com/science/article/pii/S074959781200091X.

Halszka Jarodzka, Irene Skuballa, and Hans Gruber. Eye-tracking in educational practice: Investigating visual perception underlying teaching and learning in the classroom. *Educational Psychology Review*, 33(1):1–10, 2021. ISSN 1573-336X. doi:10.1007/s10648-020-09565-7 URL https://doi.org/10.1007/s10648-020-09565-7.

R. G. Macdonald and B. W. Tatler. Gaze in a real-world social interaction: A dual eye-tracking study. *Q J Exp Psychol (Hove)*, 71(10):2162–2173, 2018. ISSN 1747-0226 (Electronic) 1747-0218 (Linking). doi:10.1177/1747021817739221 URL https://www.ncbi.nlm.nih.gov/pubmed/30226438.

Ulrich J. Pfeiffer, Kai Vogeley, and Leonhard Schilbach. From gaze cueing to dual eye-tracking: Novel approaches to investigate the neural correlates of gaze in social interaction. *Neuroscience Biobehavioral Reviews*, 37(10, Part 2):2516–2528, 2013. ISSN 0149-7634. doi:https://doi.org/10.1016/j.neubiorev.2013.07.017 URL https://www.sciencedirect.com/science/article/pii/S0149763413001887.

Matthias J. Wieser, Paul Pauli, Georg W. Alpers, and Andreas Mühlberger. Is eye to eye contact really threatening and avoided in social anxiety?—an eye-tracking and psychophysiology study. *Journal of Anxiety Disorders*, 23(1):93–103, 2009. ISSN 0887-6185. doi:https://doi.org/10.1016/j.janxdis.2008.04.004 URL https://www.sciencedirect.com/science/article/pii/S0887618508000972.

Nicolas Krucien, Mandy Ryan, and Frouke Hermens. Visual attention in multi-attributes choices: What can eye-tracking tell us? *Journal of Economic Behavior Organization*, 135:251–267, 2017. ISSN 0167-2681. doi:https://doi.org/10.1016/j.jebo.2017.01.018 URL https://www.sciencedirect.com/science/article/pii/S0167268117300264.
Luca Polonio, Sibilla Di Guida, and Giorgio Coricelli. Strategic sophistication and attention in games: An eye-tracking study. *Games and Economic Behavior*, 94:80–96, 2015. ISSN 0899-8256. doi:https://doi.org/10.1016/j.geb.2015.09.003 URL https://www.sciencedirect.com/science/article/pii/S089982561500130X.

Elena Reutskaja, Rosemarie Nagel, Colin F. Camerer, and Antonio Rangel. Search dynamics in consumer choice under time pressure: An eye-tracking study. *American Economic Review*, 101(2):900–926, 2011. doi:10.1257/aer.101.2.900 URL https://www.aeaweb.org/articles?id=10.1257/aer.101.2.900.

Ralf Van Der Lans, Rik Pieters, and Michel Wedel. Eye-movement analysis of search effectiveness. *Journal of the American Statistical Association*, 103(482):452–461, 2008. ISSN 0162-1459. doi:10.1198/016214507000000437 URL https://doi.org/10.1198/016214507000000437.

Katarzyna Harezlak and Pawel Kasprowski. Application of eye tracking in medicine: A survey, research issues and challenges. *Computerized Medical Imaging and Graphics*, 65:176–190, 2018. ISSN 0895-6111. doi:https://doi.org/10.1016/j.compmedimag.2017.04.006 URL https://www.sciencedirect.com/science/article/pii/S0895611117300435.

Ling Tao, Quan Wang, Ding Liu, Jing Wang, Ziqing Zhu, and Li Feng. Eye tracking metrics to screen and assess cognitive impairment in patients with neurological disorders. *Neurological Sciences*, 41(7):1697–1704, 2020. ISSN 1590-3478. doi:10.1007/s10072-020-04310-y URL https://doi.org/10.1007/s10072-020-04310-y.

Mélodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. Wearable eye tracking for mental health monitoring. *Computer Communications*, 35(11):1306–1311, 2012. ISSN 0140-3664. doi:https://doi.org/10.1016/j.comcom.2011.11.002 URL https://www.sciencedirect.com/science/article/pii/S0140366411003549.

A. Kullmann, R. C. Ashmore, A. Braverman, C. Mazur, H. Snapp, E. Williams, M. Szczupak, S. Murphy, K. Marshall, J. Crawford, C. D. Balaban, M. Hoffer, and A. Kideman. Portable eye-tracking as a reliable assessment of oculomotor, cognitive and reaction time function: Normative data for 18-45 year old. *PLoS One*, 16(11):e0260351, 2021. ISSN 1932-6203 (Electronic) 1932-6203 (Linking). doi:10.1371/journal.pone.0260351 URL https://www.ncbi.nlm.nih.gov/pubmed/34807938.

E. F. Fincham and J. Walton. The reciprocal actions of accommodation and convergence. *J Physiol*, 137(3):488–508, 1957. ISSN 0022-3751 (Print) 0022-3751 (Linking). doi:10.1113/jphysiol.1957.sp005829 URL https://www.ncbi.nlm.nih.gov/pubmed/13463783.

Andrew J. Parker. Binocular depth perception and the cerebral cortex. *Nature Reviews Neuroscience*, 8(5):379–391, 2007. ISSN 1471-0048. doi:10.1038/nrn2131 URL https://doi.org/10.1038/nrn2131.

K. Preuschhoff, B. M. t Hart, and W. Einhauser. Pupil dilation signals surprise: Evidence for noradrenaline’s role in decision making. *Front Neurosci*, 5:115, 2011. ISSN 1662-453X (Electronic) 1662-453X (Linking). doi:10.3389/fnins.2011.00115 URL https://www.ncbi.nlm.nih.gov/pubmed/21994487.

M. Sole Puig, L. Perez Zapata, J. A. Aznar-Casanova, and H. Super. A role of eye vergence in covert attention. *PLoS One*, 8(1):e52955, 2013. ISSN 1932-6203 (Electronic) 1932-6203 (Linking). doi:10.1371/journal.pone.0052955 URL https://www.ncbi.nlm.nih.gov/pubmed/23382827.

David M. Hoffman, Ahna R. Girshick, Kurt Akeley, and Martin S. Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3):33–33, 2008. ISSN 1534-7362. doi:10.1167/8.3.33 URL https://doi.org/10.1167/8.3.33.

A. Gibaldi and M. S. Banks. Binocular eye movements are adapted to the natural environment. *J Neurosci*, 39(15):2877–2888, 2019. ISSN 1529-2401 (Electronic) 0270-6474 (Linking). doi:10.1523/JNEUROSCI.2591-18.2018 URL https://www.ncbi.nlm.nih.gov/pubmed/30733219.
A. Rufa, G. L. Mariottini, D. Prattichizzo, D. Alessandrini, A. Vicino, and A. Federico. Video-based eye tracking: our experience with advanced stimuli design for eye tracking software. Ann N Y Acad Sci, 1039:575–9, 2005. ISSN 0077-8923 (Print) 0077-8923 (Linking). doi:10.1196/annals.1325.071 URL https://www.ncbi.nlm.nih.gov/pubmed/15827026

L. H. Yu and M. Eizenman. A new methodology for determining point-of-gaze in head-mounted eye tracking systems. IEEE Trans Biomed Eng, 51(10):1765–73, 2004. ISSN 0018-9294 (Print) 0018-9294 (Linking). doi:10.1109/TBME.2004.831523 URL https://www.ncbi.nlm.nih.gov/pubmed/15490823

Matteo Cognolato, Manfredo Atzori, and Henning Müller. Head-mounted eye gaze tracking devices: An overview of modern devices and recent advances. Journal of Rehabilitation and Assistive Technologies Engineering, 5:2055668318773991, 2018. ISSN 2055-6683. doi:10.1177/2055668318773991 URL https://doi.org/10.1177/2055668318773991

Ji Woo Lee, Chul Woo Cho, Kwang Yong Shin, Eui Chul Lee, and Kang Ryong Park. 3d gaze tracking method using purkinje images on eye optical model and pupil. Optics and Lasers in Engineering, 50(5):736–751, 2012. ISSN 0143-8166. doi:https://doi.org/10.1016/j.optlaseng.2011.12.001 URL https://www.sciencedirect.com/science/article/pii/S0143816611003460

Nicholas Wade and Benjamin Tatler. The Moving Tablet of the Eye: The Origins of Modern Eye Movement Research. Oxford University Press, 2005.

S. Kim, M. Sked, and Q. Ji. Non-intrusive eye gaze tracking under natural head movements. Conf Proc IEEE Eng Med Biol Soc, 2004:2271–4, 2004. ISSN 1557-170X (Print) 1557-170X (Linking). doi:10.1109/IEMBS.2004.1403660 URL https://www.ncbi.nlm.nih.gov/pubmed/17272180

Stephan Wenger, Miska M Hannuksela, Thomas Stockhammer, M Westerlund, and D Singer. Rtp payload format for h.264 video. Report 2070-1721, 2005.

J. Hwang, A. R. Mitz, and E. A. Murray. Nimh monkeylogic: Behavioral control and data acquisition in matlab. J Neurosci Methods, 323:13–21, 2019. ISSN 0165-0270 (Print) 0165-0270. doi:10.1016/j.jneumeth.2019.05.002

Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000. ISSN 1939-3539. doi:10.1109/34.888718

E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: A hands-on survey. IEEE Trans Vis Comput Graph, 22(12):2633–2651, 2016. ISSN 1077-2626. doi:10.1109/tvcg.2015.2513408

Charlie S. Burlingham and David J. Heeger. Heading perception depends on time-varying evolution of optic flow. Proceedings of the National Academy of Sciences, 117(52):33161–33169, 2020. doi:10.1073/pnas.202984117 URL https://doi.org/10.1073/pnas.202984117

Siwen Guo, Jin Wu, Zuocai Wang, and Jide Qian. Novel marg-sensor orientation estimation algorithm using fast kalman filter. Journal of Sensors, 2017:8542153, 2017. ISSN 1687-725X. doi:10.1155/2017/8542153 URL https://doi.org/10.1155/2017/8542153

S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In 2011 IEEE International Conference on Rehabilitation Robotics, pages 1–7. ISBN 1945-7901. doi:10.1109/ICORR.2011.5975346

R. Ramachandran and S. G. Lisberger. Normal performance and expression of learning in the vestibulo-ocular reflex (vor) at high frequencies. J Neurophysiol, 93(4):2028–38, 2005. ISSN 0022-3077 (Print) 0022-3077. doi:10.1152/jn.00832.2004

A. Cova and H. L. Galiana. A bilateral model integrating vergence and the vestibulo-ocular reflex. Experimental Brain Research, 107(3):435–452, 1996. ISSN 1432-1106. doi:10.1007/BF00230424 URL https://doi.org/10.1007/BF00230424
Jasper H. Fabius, Alessio Fracasso, and Stefan Van der Stigchel. Spatiotopic updating facilitates perception immediately after saccades. *Scientific Reports*, 6(1):34488, 2016. ISSN 2045-2322. doi:10.1038/srep34488 URL https://doi.org/10.1038/srep34488

Jason M. Samonds, Wilson S. Geisler, and Nicholas J. Priebe. Natural image and receptive field statistics predict saccade sizes. *Nature Neuroscience*, 21(11):1591–1599, 2018. ISSN 1546-1726. doi:10.1038/s41593-018-0255-5 URL https://doi.org/10.1038/s41593-018-0255-5

Hamed Rahimi-Nasrabadi, Jianzhong Jin, Reece Mazade, Carmen Pons, Sohrab Najafian, and Jose-Manuel Alonso. Image luminance changes contrast sensitivity in visual cortex. *Cell Reports*, 34(5):108692, 2021. ISSN 2211-1247. doi:https://doi.org/10.1016/j.celrep.2021.108692. URL https://www.sciencedirect.com/science/article/pii/S221112472100005X

S. Trauzettel-Klosinski. Current methods of visual rehabilitation. *Dtsch Arztebl Int*, 108(51-52):871–8, 2011. ISSN 1866-0452. doi:10.3238/arztebl.2011.0871

Carlos J. Hernández-Rodríguez, David P. Piñero, Ainhoa Molina-Martín, León Morales-Quezada, Dolores de Fez, Luis Leal-Vega, Juan F. Arenillas, and María C. Coco-Martín. Stimuli characteristics and psychophysical requirements for visual training in amblyopia: A narrative review. *Journal of Clinical Medicine*, 9(12), 2020. ISSN 2077-0383. doi:10.3390/jcm9123985

Guy Bouvier, Yuta Senzai, and Massimo Scanziani. Head movements control the activity of primary visual cortex in a luminance-dependent manner. *Neuron*, 108(3):500–511.e5, 2020. ISSN 0896-6273. doi:https://doi.org/10.1016/j.neuron.2020.07.004 URL https://www.sciencedirect.com/science/article/pii/S0896627320305213

Matthias Nau, Joshua B. Julian, and Christian F. Doeller. How the brain’s navigation system shapes our visual experience. *Trends in Cognitive Sciences*, 22(9):810–825, 2018. ISSN 1364-6613. doi:https://doi.org/10.1016/j.tics.2018.06.008 URL https://www.sciencedirect.com/science/article/pii/S1364661318301487

Hiroshi M. Shiozaki and Hokto Kazama. Parallel encoding of recent visual experience and self-motion during navigation in drosophila. *Nature Neuroscience*, 20(10):1395–1403, 2017. ISSN 1546-1726. doi:10.1038/nn.4628 URL https://doi.org/10.1038/nn.4628

Mateo Vélez-Fort, Edward F. Bracey, Sepiedeh Keshavarzi, Charly V. Rousseau, Lee Cossell, Stephen C. Lenz, Molly Strom, and Troy W. Margrie. A circuit for integration of head- and visual-motion signals in layer 6 of mouse primary visual cortex. *Neuron*, 98(1):179–191.e6, 2018. ISSN 0896-6273. doi:https://doi.org/10.1016/j.neuron.2018.02.023 URL https://www.sciencedirect.com/science/article/pii/S0896627318301442

W. Einhauser, F. Schumann, S. Bardins, K. Bartl, G. Boning, E. Schneider, and P. König. Human eye-head coordination in natural exploration. *Network*, 18(3):267–97, 2007. ISSN 0954-898X (Print) 0954-898X (Linking). doi:10.1080/09548980701671094 URL https://www.ncbi.nlm.nih.gov/pubmed/17926195

Otto Lappi. Eye movements in the wild: Oculomotor control, gaze behavior frames of reference. *Neuroscience Biobehavioral Reviews*, 69:49–68, 2016. ISSN 0149-7634. doi:https://doi.org/10.1016/j.neubiorev.2016.06.006 URL https://www.sciencedirect.com/science/article/pii/S0149763415301317

Mary Hayhoe and Dana Ballard. Eye movements in natural behavior. *Trends in Cognitive Sciences*, 9(4):188–194, 2005. ISSN 1364-6613. doi:https://doi.org/10.1016/j.tics.2005.02.009 URL https://www.sciencedirect.com/science/article/pii/S1364661305000598

Ahmad F. Klaib, Nawaf O. Alsrehin, Wasen Y. Melhem, Haneen O. Bashtawi, and Awas A. Magableh. Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and internet of things technologies. *Expert Systems with Applications*, 166:114037, 2021. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2020.114037 URL https://www.sciencedirect.com/science/article/pii/S0957417420306071
Konstantinos-Filippos Kollias, Christine K. Syriopoulou-Delli, Panagiotis Sarigiannidis, and George F. Fragulis. The contribution of machine learning and eye-tracking technology in autism spectrum disorder research: A systematic review. *Electronics*, 10(23), 2021. ISSN 2079-9292. doi:10.3390/electronics10232982.

Raimondas Zemblys, Diederick C. Niehorster, Oleg Komogortsev, and Kenneth Holmqvist. Using machine learning to detect events in eye-tracking data. *Behavior Research Methods*, 50(1):160–181, 2018. ISSN 1554-3528. doi:10.3758/s13428-017-0860-3. URL https://doi.org/10.3758/s13428-017-0860-3.

Ben Mildenhall, Peter Hedman, Ricardo Martín-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199.

C. Lee, A. K. Kosta, and K. Roy. Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures. In 2022 *International Conference on Robotics and Automation (ICRA)*, pages 6504–6510. doi:10.1109/ICRA46639.2022.9811821.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 477–490. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/srivastava22a.html.

C. Gan, S. Zhou, J. Schwartz, S. Alter, A. Bhandwad, D. Gutfreund, D. L. K. Yamins, J. J. DiCarlo, J. McDermott, A. Torralba, and J. B. Tenenbaum. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. In 2022 *International Conference on Robotics and Automation (ICRA)*, pages 8847–8854. doi:10.1109/ICRA46639.2022.9812329.