nature protocols

PROTOCOL

https://doi.org/10.1038/s41596-018-0002-0

Corrected: Publisher Correction

Evoking and tracking zebrafish eye movement in multiple larvae with ZebEyeTrack

Florian A. Dehmelt[1], Adam von Daranyi[2], Claire Leyden[1,3] and Aristides B. Arrenberg[1*]

Reliable measurement of spontaneous and evoked eye movement is critical for behavioral vision research. Zebrafish are increasingly used as a model organism for visual neural circuits, but ready-to-use eye-tracking solutions are scarce. Here, we present a protocol for automated real-time measurement of angular horizontal eye position in up to six immobilized larval fish using a custom-built LabVIEW-based software, ZebEyeTrack. We provide its customizable source code, as well as a streamlined and compiled version, ZebEyeTrack Light. The full version of ZebEyeTrack controls all required hardware and synchronizes six essential aspects of the experiment: (i) stimulus design; (ii) visual stimulation with moving bars; (iii) eye detection and tracking, as well as general motion detection; (iv) real-time analysis; (v) eye-position-dependent closed-loop event control; and (vi) recording of external event times. This includes optional integration with external hardware such as lasers and scanning microscopes. Once installation is complete, experiments, including stimulus design, can be completed in <10 min, and recordings can last anywhere between seconds and many hours. Results include digitized angular eye positions and hardware status, which can be used to compute tuning curves, optokinetic gain, and other custom data analysis. After the experiment, or based on existing videos, optokinetic response (OKR) performance can be analyzed semi-automatically via the graphical user interface, and results can be exported. ZebEyeTrack has been used successfully for psychophysics experiments, for optogenetic stimulation, and in combination with calcium imaging.

Introduction

Understanding the neural circuitry linking visual input to motor output requires quantitative knowledge of both. Eye movements are of special interest because of their direct involvement in many visually mediated behaviors. In the laboratory, they can either occur spontaneously or be evoked by the experimenter to study the relation between a specific stimulus and behavior. A common reflex across vertebrate species is the OKR, a gaze-stabilization behavior, which is readily evoked by presenting a pattern of moving vertical bars. The OKR consists of slow-phase (following) and quick-phase (saccadic) eye movements to minimize retinal slip. The OKR is widely used to measure visual function in human subjects as well as in model organisms, including rodents[1–3], teleost fish[4–9], and non-human primates[10,11].

In larval and adult zebrafish, the OKR has been exploited in forward genetic screens to identify mutations affecting the visual system[7,8,12,13]. More recently, oculomotor behavior has been quantified to investigate the neural underpinnings of visually or vestibularly mediated behaviors and visual function[14–17]. Most of these studies used either a commercial set of software and hardware[18,19] or custom-built, unpublished software tailored to a specific setup[12,15,16]. This includes several studies using unpublished precursor versions of our software[14,17,20–25]. For simpler experiments, manual inspection may be sufficient[13].The commercial solution is highly convenient for a narrow range of applications such as genetic screens. But by design, it cannot accommodate more complex experimental paradigms involving a variety of stimuli, optogenetic modulation, parallel quantification of tail and body motion, two-photon microscopy, or other pre-existing optical setups. Many existing solutions also cannot track the eyes of multiple fish in parallel. In this protocol, we describe the automated real-time measurement of angular horizontal eye position in up to six immobilized larval zebrafish using a custom-built LabVIEW-based software, ZebEyeTrack, and discuss how to integrate it into a variety of experimental paradigms.

[1]Werner Reichardt Centre for Integrative Neuroscience and Institute of Neurobiology, University of Tübingen, Tübingen, Germany. [2]Werner Reichardt Centre for Integrative Neuroscience, Central Office System Administration, University of Tübingen, Tübingen, Germany. [3]Graduate Training Centre of Neuroscience, University of Tübingen, Tübingen, Germany. *e-mail: aristides.arrenberg@uni-tuebingen.de

Box 1 | Eye tracking for an existing video ● Timing 5 min + length of the video

Procedure

1 Verify that the video is a compressed or uncompressed 8-bit .avi file (e.g., Supplementary Video 1, the uncompressed video is available via http://www.zebeyetrack.org/videos/video_1.avi).
▲ **CRITICAL STEP** ZebEyeTrack fully supports compressed .avi files. However, some types of compression require users to install specific codecs, and others may reduce tracking precision, so uncompressed video is preferred.
2 To execute virtual instrument (VI), click on the white arrow button in the upper-left corner of the LabVIEW window. If the arrow is black, the program is already running.
▲ **CRITICAL STEP** If buttons on the settings tab are grayed out, the software is already running in a particular operating mode. In this case, click 'Stop', and then click the white arrow again to restart.
▲ **CRITICAL STEP** Select 'Run Continuously' (circular arrows) to quickly restart the program when analyzing multiple files.
3 Select 'Path and *.txt file name' using the free-text field. This name will be the root name for all files created during operation (eye-tracking data, video, post hoc analysis, and body axes). It can be changed again during operation. If the file name is not changed between recordings, new data will be appended to the bottom of the existing data in the file.
▲ **CRITICAL STEP** Choose a file name with the appropriate file-type extension, .txt. Otherwise, the file name may be truncated.
4 Click on 'analyse existing *.avi file', then click on 'accept these settings and run programme'.
5 Once the program has started, a pop-up window appears, allowing the user to browse the local hard drive. Find and select the file you wish to process. To get started, use the video file provided in supplementary material (Supplementary Video 1).
6 On the 'Eye detection & recording' panel, the video should play in an infinite loop. For large videos, disable the 'display' button to increase processing speed.
7 (Optional) At any time between recordings, the file path can be modified by clicking on 'Update data file path' on the 'Eye detection & recording' panel. ZebEyeTrack Light will create a new folder if it does not exist. Click on 'restore previous' to fill in the name of the last file recorded.
8 Follow steps 14–19 of the main Procedure to identify body axes and debug eye detection.
9 To start data acquisition, click on 'track eye position' on the 'Eye detection & recording' tab. The infinite video loop will stop, the video will start over again, and it will run a single time. Data acquisition will end automatically at the end of the video file or whenever 'track eye position' is manually deactivated.
10 (Optional) While the video is running, it can be compared with the real-time eye-tracking readout at the bottom of the screen. Right-click icons on the right and select 'Plot visible' to toggle the visibility of individual graphs. In case you notice a mismatch, refer to the Troubleshooting section. For longer recordings exceeding the length of the bottom panel, follow steps 31–33 of the main Procedure.
11 ZebEyeTrack automatically writes eye-tracking data to a tab-separated .txt file. Its file name is the one chosen in Step 7 or 13 of the main Procedure, and its content is explained in Table 1. Once data are written, infinite playback of the video resumes.

Overview of the procedure

To perform an oculomotor experiment with ZebEyeTrack, hardware and software must be installed and tested (Steps 1–4). Live zebrafish larvae must be prepared and immobilized in agarose, in accordance with all relevant institutional and legal regulations (Step 5). Because ZebEyeTrack supports many different types of experiments, it must be configured upon launch to activate the required features and deactivate others for improved performance (Steps 6–13). Once running, the location and orientation of fish must be confirmed manually, and automatic eye detection and tracking begins (Steps 14–19). If desired, analog output can be scheduled or generated manually, e.g., for optogenetics (Steps 20 and 21). Eyes are tracked and all data are saved to .txt files (Steps 22–24). If desired, visual stimuli can be designed, saved, loaded, and presented in four different ways (Steps 25–28). If desired, live video can be recorded (Steps 29 and 30), and previous or ongoing recordings can be inspected (Steps 31–33). Data files are automatically saved during an experiment (Step 34). Finally, users can analyze eye-tracking data (Table 1) semi-automatically via the ZebEyeTrack graphical user interface (GUI) (Step 35). Beyond providing live analysis of experiments, ZebEyeTrack and ZebEyeTrack Light can analyze existing videos of zebrafish larvae, including those acquired using other protocols (Box 1). Users can add custom procedures to drive additional hardware, provide additional stimuli, or analyze data in real time by editing the available source code. As an example, we incorporated general motion detection (Box 2, Supplementary Fig. 1) without affecting other parts of the software. Once hardware has been installed and configured, preparatory steps of the experiment, including stimulus design, can be completed in under 10 min, and recordings can last anywhere between seconds and many hours. This, as well as the possibility of customizing the source code, makes it straightforward to integrate ZebEyeTrack into new or existing experimental procedures. For added convenience and prior to installing ZebEyeTrack on a new computer, interested users can test the software at zebeyetrack.com. A virtual machine has been set up there, which also enables users to analyze their own data (see instructions at zebeyetrack.com).

Box 2 | General motion detection ● Timing 2 min

Procedure

1 To select an ROI, navigate to the 'Motion detection' panel. Click on 'display video' on the top left and draw any shape of ROI using the tools next to the display. Deactivate 'display video' later on to improve performance.

2 To inspect the ROI, click on 'display video' on the top right to see a live image of the pixels selected. Adjust the ROI if needed. Deactivate 'display video' later on to improve performance.

▲ **CRITICAL STEP** Non-rectangular ROIs are padded with zero-value (black) pixels to form a rectangle. These will lower the mean pixel intensity, so you may need to set lower thresholds for non-rectangular ROIs.

3 To inspect the pixel intensity time course, click on 'show graphs' to display the mean ROI pixel intensity and its frame-to-frame rate of change. The currently chosen threshold is also displayed. Because the rate of change is usually much lower than the absolute value of pixel intensity, it is displayed at tenfold magnitude for improved visibility, alongside a second threshold line, also at tenfold magnitude. Compare the intensity curve to the lower (1×) threshold line to identify a plausible (1×) intensity threshold, and compare the rate-of-change curve to the upper (10×) threshold line to select an appropriate (10×) rate-of-change threshold.

4 Set the 'threshold' to the value desired. Select one of the radio buttons to choose whether mean pixel intensity or its rate of change will generate a trigger whenever they cross the user-defined thresholds. Select whether a trigger should be generated only for upward, only for downward, or for any kind of threshold crossing.

! CAUTION Trigger generation begins immediately. Exercise caution when triggering potentially harmful output such as optogenetic stimulation.

5 By default, ROI pixel intensities are not saved to the eye-tracking data file. Thus, they are not listed in any of the relevant tables (Table 1, Supplementary Table 1). To save them to an additional, second-to-last column of the file, click on 'include ROI mean in main .txt data file'. Then navigate to the 'Eye detection & recording tab' to start a recording, as described in Step 22 of the main Procedure.

▲ **CRITICAL STEP** To avoid an accidental change of data file format, this option is automatically deactivated at each start of the program. Reactivate if needed.

Software capabilities

Most approaches for investigating the neural and behavioral response of animals to sensory stimuli share as a common challenge the fact that experimenters need to control a wide range of hardware to run a stimulus protocol, acquire data, and analyze it. Often, this requires commercial software or forces researchers to develop their own code from scratch. Although commercial solutions offer advantages with respect to comfort and technical support, they cannot readily be customized to the specific needs of each user, and the associated cost discourages researchers from exploring new experimental paradigms.

Software and hardware requirements

We seek to facilitate the development of novel protocols involving zebrafish eye tracking by making our software freely available. Although ZebEyeTrack itself is freeware, running the full version requires some commercial software: National Instruments' LabVIEW, the NI Vision Development Module, and MATLAB, including its freeware, Psychtoolbox ('Equipment' section). To present stimuli and acquire data, computer screens, video projectors, or LED arrays, as well as a data acquisition (DAQ) device and a camera are required. As an alternative, we provide a light version of our software, ZebEyeTrack Light, in which the code is compiled and for which no specific hardware is required, limiting the cost to that for MATLAB and the NI Vision Runtime Engine alone (of which free trial versions are available).

ZebEyeTrack versus ZebEyeTrack Light

The reduced version, ZebEyeTrack Light, is limited to five specific functions: recording of video data, tracking of eyes in existing or live video data, performance of post hoc analysis on these traces, presentation of visual stimuli via custom LED arrays, and generation of analog output (Fig. 1). In addition, the full version ZebEyeTrack can present visual stimuli via computer displays and generate other outputs, such as optogenetic stimulation, with optional event-controlled triggering (Figs. 1 and 2). It also enables post hoc synchronization with experimental hardware that cannot itself be driven by ZebEyeTrack or for which users simply prefer the existing proprietary software. To perform synchronization, an analog signal representing the time course of this hardware (e.g., the voltage signal driving a microscope-scanning mirror) is combined with the entire set of eye-tracking data into a single data file with shared time stamps. In parallel, videos can be recorded as .avi files. While eye-tracking data are acquired, the software performs real-time analysis detecting and counting saccades. Once eye tracking is complete, post hoc analysis is available through the user interfaces of both software versions. All data are combined into a single .txt file for further custom data analysis.

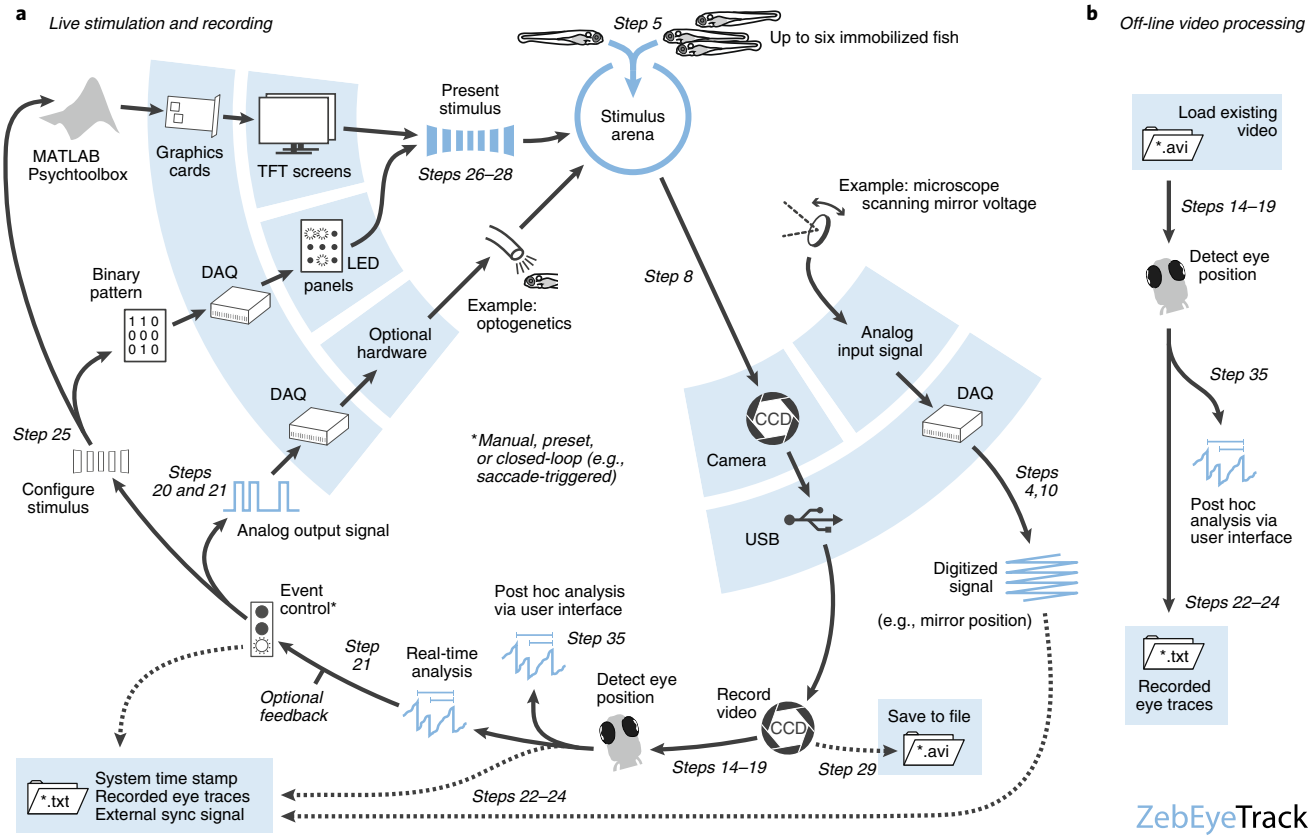

Fig. 1 | ZebEyeTrack capabilities. a,b, ZebEyeTrack is a modular software able to control multiple aspects of an oculomotor experiment, from stimulus presentation and data acquisition to eye tracking and data analysis (**a**), or to simply track the eyes in an existing video of larval zebrafish (**b**). In both cases, the eyes of up to six fish can be tracked in parallel. As shown in **a**, users can choose to control several alternative setups and optional hardware. Stimuli can be presented on computer screens or custom LED arrays. Computer screens can be replaced by video projection. Video data can be recorded and saved to a file. Eyes are automatically detected and their positions are tracked. Analog output can be generated to provide eye-movement-dependent event control for optogenetics or other application-specific hardware. Analog and digital input from external hardware can be acquired and saved to a single spreadsheet file, along with detailed eye-tracking data, for custom post hoc data analysis. This includes mirror positions for scanning microscopes. All functions illustrated here are controlled directly through the ZebEyeTrack graphical user interface, which also offers basic data-analysis tools. They can be modified at will by editing the freely available LabVIEW and MATLAB source code. *Event control can be manual, scheduled, or closed-loop using saccade triggers. Numbers indicate corresponding steps of the main Procedure.

Comparison with other methods

Several alternative solutions for zebrafish eye tracking exist. Most often, the goal behind obtaining eye traces from video data is to quantify OKR parameters such as saccade frequency, and the gain and velocity of slow phase-eye movements between saccades. Arguably, the simplest approach is visual inspection by a human observer[26]. Eye positions can also be drawn manually using free image-analysis software such as ImageJ. For added convenience, a custom ImageJ plugin has been used to identify weak OKR phenotypes[12]. Finally, bundles of commercial hardware and software[18,19,27] are distributed by TSE Systems and ViewPoint (Table 2, nos. 1 and 2). Numerous other solutions exist to quantify related behaviors in other species, such as optomotor responses in mice. Often, these use other variables such as head direction to infer gaze and do not measure eye position[28]. Even if they do, the distinct shape and contrast of zebrafish eyes require distinct detection algorithms.

ZebEyeTrack is specifically designed for studying zebrafish OKR. It is much more efficient than manual analysis, as many hours of video recordings can be processed automatically once eye detection has been manually confirmed just once. As opposed to mere image-analysis software, ZebEyeTrack incorporates additional data such as the time course of other experimental variables. In addition, it offers a variety of ways to control experiments in real time instead of just analyzing their outcomes. The cost of ZebEyeTrack is limited to a number of supporting software licenses, all of which are available as free trial versions; and ZebEyeTrack Light can be used indefinitely after a small investment in MATLAB and the NI Vision Runtime Engine. Modular and customizable, ZebEyeTrack integrates

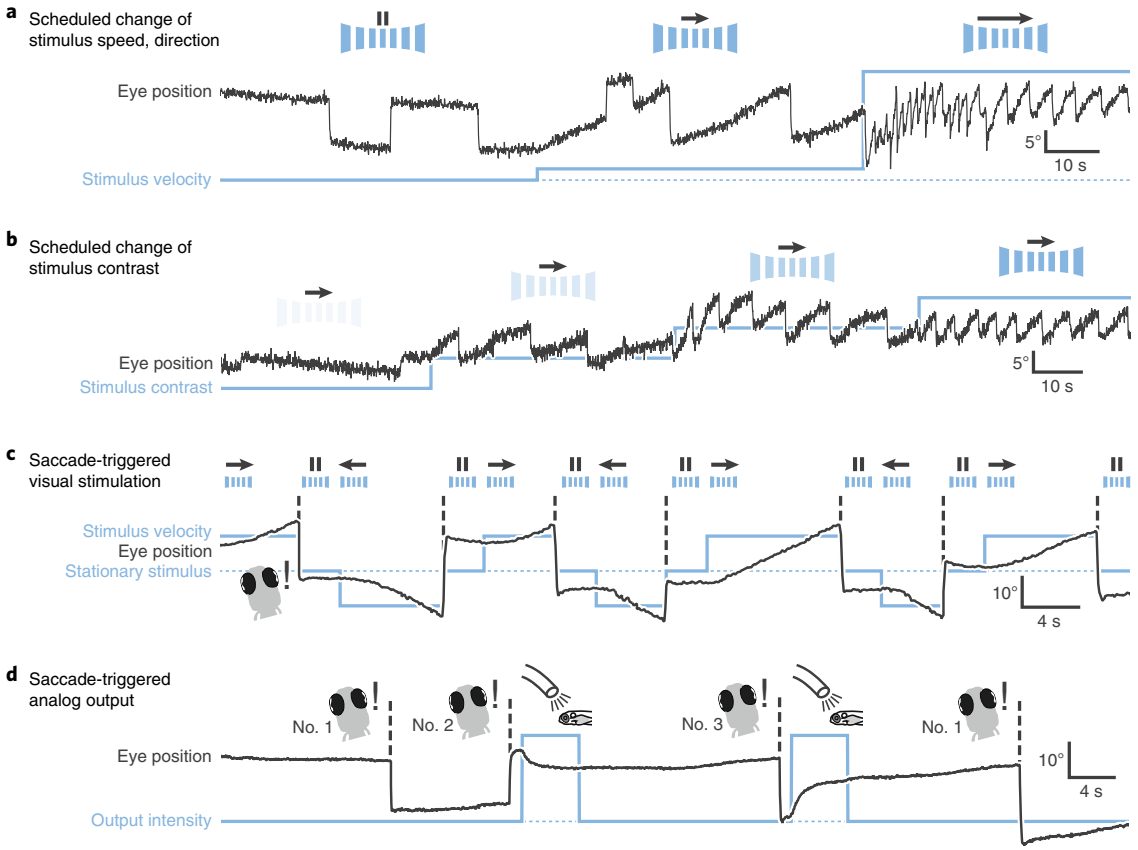

Fig. 2 | Eye-tracking data are acquired for a variety of experimental paradigms. a, On computer screens, projection screens, or custom LED arrays, ZebEyeTrack can present preprogrammed sequences of horizontally moving vertical bars and detect the resulting angular eye position of zebrafish larvae (dark gray). The gain of OKR eye movement in each stimulus phase depends on the stimulus parameters of the phase. Here, preprogrammed stimuli of different angular velocity (0, 1.41, and 45.0 degrees/s; light blue) are presented. Other stimulus parameters, such as spatial frequency, direction of motion, and duration, can also be altered but were kept constant in this example. **b**, On computer and projection screens, but not custom LED arrays, stimulus contrast can also be adjusted. **c**,**d**, ZebEyeTrack also offers two closed-loop modes with automated saccade detection. **c**, First mode: using custom LED arrays, stimulus motion can be interrupted for a user-defined duration after each saccade and then resumed in either the same or the opposite direction (light blue). **d**, Second mode: analog output (light blue) with custom voltage, duration, and post-saccade delay can be generated whenever a saccade is detected. This can be used to drive optogenetic stimulation, such as that for halorhodopsin-mediated inhibition of oculomotor integrator neurons (shown here). As a built-in control, one in three saccades (marked No. 1) does not elicit analog output. Where shown, the dashed blue line indicates a baseline value of zero. Appropriate regulatory board permission was obtained before zebrafish experiments.

easily into existing and novel setups, whereas commercial all-in-one products are constrained to convenient, but narrowly defined, applications. Under a Creative Commons license, users may adapt the source code of ZebEyeTrack to integrate their own custom hardware, modify visual stimuli, include other modalities such as auditory stimulation, and expand the existing event control options to suit their needs.

It is worth noting that freeware environments such as Bonsai[39], in principle, allow for the creation of comparable software, but neither we nor the developers of this framework are aware of any currently existing Bonsai applications with a similar purpose (G. Lopes, personal communication). Although ZebEyeTrack is a ready-to-use software application for eye tracking, Bonsai, by contrast, is a general-purpose programming environment.

Limitations

With our own setup, we achieve a maximum sampling rate of ~30 frames per second (f.p.s.). This could be improved by relying on a more powerful computer, but it is already more than sufficient for most applications. Furthermore, the inherent design of the software, which tries to run through the loop iterations as fast as it can, results in the sampling time not being the same for every sample. This

information is retained, as all time stamps are saved to the data file, but might cause difficulties in certain applications.

Applications

ZebEyeTrack has been tried and tested for the following range of applications.

Spontaneous eye movements

Spontaneous eye movements in either high-contrast or low-contrast conditions can be tracked and quantified to count the number of saccades[22,29].

Evoked eye movements

The OKR can be evoked by mono- or binocular presentation of moving stripes of adjustable contrast, spatial frequency, and speed according to the designed stimulus protocol. A typical protocol would run different stimulus conditions (stimulus protocol phases) to establish behavioral tuning curves, e.g., for contrast. Saccade number and slow phase velocity can be automatically quantified for each stimulus condition after the experiment. In conjunction with photoconversion experiments, we used ZebEyeTrack to localize saccade-generating neurons in the zebrafish hindbrain and characterized visual acuity and contrast sensitivity in mutants[14].

Event control for optogenetics

ZebEyeTrack offers active event control during an experiment. Its universal analog voltage output signal can be used to control an analog-modulated laser or LED for optogenetic stimulation, or to trigger additional stimuli (visual or otherwise). Because saccades are detected online, the program can be run in a closed-loop setting, in which each saccade triggers optogenetic stimulation[20,22]. For example, we applied optogenetic perturbations after each saccade to localize the position of the hindbrain oculomotor integrator[23], and combined such optogenetic stimulations with computational modeling to characterize the dynamics of the hindbrain oculomotor integrator[22]. In addition, Thiele et al. used a previous version of the software to synchronize optogenetic stimulation, visual stimulation, and imaging[25].

Calcium imaging

ZebEyeTrack can be used in calcium imaging experiments to present visual stimuli, record eye positions, and take time stamps for these variables, as well as for an additionally sampled analog voltage input channel. A signal from the microscope (e.g., the y-scanning mirror voltage) can be fed into this channel for post hoc synchronization of stimulus and behavioral data on any scanning microscope (e.g., confocal microscope or two-photon microscope). In a two-photon scanning setup, interference of stimulus light with fluorescence detection can be abolished by using a visual stimulus arena consisting of LED panels that can be switched on and off very fast (<1 μs). For this purpose, we developed an LED arena in which a line scan signal from the microscope controls LED irradiation in such a way that the visual stimulus is displayed only during the return period of the microscope x-scanning mirror[17]. The stimulus pattern of this LED arena is controlled by ZebEyeTrack's eight digital out signals. We used this ZebEyeTrack functionality for simultaneous eye tracking and two-photon calcium imaging during visual stimulation to determine single-cell tuning to stimulus temporal frequency and stimulus direction[17]. In these experiments, laser scanning and microscope image acquisition were controlled by commercial software running independently, and post hoc synchronization was achieved by recording time stamps with ZebEyeTrack. Most recently, we recorded the activity of dopaminergic neurons and larval behavior following tactile stimulation using a modified version of the software[24].

Event control for calcium imaging

Owing to the correlative nature of findings through calcium imaging, it is advisable to use stimulus protocols that help to temporally decorrelate the relevant variables. For example, saccade times, eye position, and eye velocity are partly correlated during OKR behavior. The event control capabilities of ZebEyeTrack include closed-loop feedback, which can be used to stop stimulus motion for several seconds immediately after a saccade is detected. This way, saccade-related motor activity in the brain is temporally separated from activity related to slow eye movements or eye position, and can reliably be distinguished in calcium imaging experiments. We are currently using this to investigate neuronal

tuning to eye movement motor variables in hindbrain structures such as the nucleus abducens and the neural integrator for horizontal eye movements[30].

Student training

Data acquisition and analysis via the GUI are straightforward, and we have successfully used them in the practical training of graduate students learning to perform simple psychophysics experiments (e.g., in a practical block course for students of the Master of Science in Neurobiology programme at the University of Tübingen). After a 20-min introduction, students were able to use ZebEyeTrack to design and present complex series of stimuli, record eye traces, and export data to .txt files without further assistance.

Experimental design

In a typical experiment, fully pigmented wild-type or minimally pigmented *mitfa*$^{-/-}$ larvae are first immobilized in agarose with their eyes cut free. Second, custom-grating stimulus protocols of different length and complexity are imported into ZebEyeTrack or designed directly through the interface. Third, these bar-shaped patterns are displayed in a stimulus arena made of computer screens (Fig. 3), video projectors (not shown), or custom LED arrays (Supplementary Fig. 2). Fourth, animal behavior is quantified simultaneously and in real time through automated eye detection and tracking (Fig. 4), using a CCD camera and a custom particle-analysis algorithm explained below (Figs. 5 and 6). Fifth, in an optional closed-loop mode, this real-time analysis is used to trigger stimulus adaptation based on the occurrence of saccades, generate a universal analog voltage signal (e.g., for modulation of neural activity using optogenetics), or trigger any other experiment-specific hardware chosen by the experimenter. Data from previous recordings can be analyzed, and the results can be saved to .txt files, directly through the GUI (Fig. 7). Each of these features can be deactivated if experimenters need only the basic functionality or wish to combine our software with some of their own solutions.

Customization of software features (optional)

As an example for possible source code customization, we have included a rudimentary tail motion detection feature, which additionally triggers analog output such as optogenetic stimulation at each tail motion event (Box 2). It is contained on a separate tab of the GUI, and users can add other such tabs for their own customizations. Our LabVIEW software, like many others, is divided into a main virtual instrument (VI) contained in a .vi file and subfunctions in separate files, referred to as sub-VIs. Users wishing to include their own custom features can integrate them by placing them in the appropriate part of the LabVIEW block diagram of the main .vi file: e.g., new features pertaining to the quantification of motion in the image can be placed in the same functional section as our tail motion detection code to ensure proper execution, and new features related to stimulus presentation should be placed within or called from our stimulus-related sub-VIs, such as VisualStimulation.vi. To link new and existing features with minimal effort, additional global variables can be created in GlobalVariables.vi.

Illumination and eye-tracking algorithm

To track the eyes, ZebEyeTrack performs particle analysis on the video frames, and a few parameters must be adjusted by the experimenter to optimize the performance of the eye-tracking algorithm. Eyes are identified by a custom particle-detection algorithm (Fig. 5) as those contiguous clusters of dark pixels that fall within the user-defined size range and are closest to the user-defined midpoint between the eyes. In this way, eye position and orientation are determined on each video frame individually. Specifically, each frame is a grayscale image, on which our custom algorithm detects those pixels darker than a given threshold. Users can and should update this threshold before acquiring data (Step 15). The default size range for particles to be tracked as eyes should work well for most applications, but it can be adjusted in the source code. Of all the particles that fall into the plausible size range, only those two closest to the midpoint between the eyes of the user-defined body axis are kept. This body axis is manually set by drawing one rostrocaudal line for each fish ahead of the recording (Step 17), and care should be taken to always begin this line directly between the eyes to avoid detection artifacts. The caudal end of the line is less critical, as long as it is placed on the midline of the animal. Angular eye position is perpendicular to the maximum Feret distance of each particle, and is expressed relative to body orientation, such that zero degrees correspond to a perfectly lateral

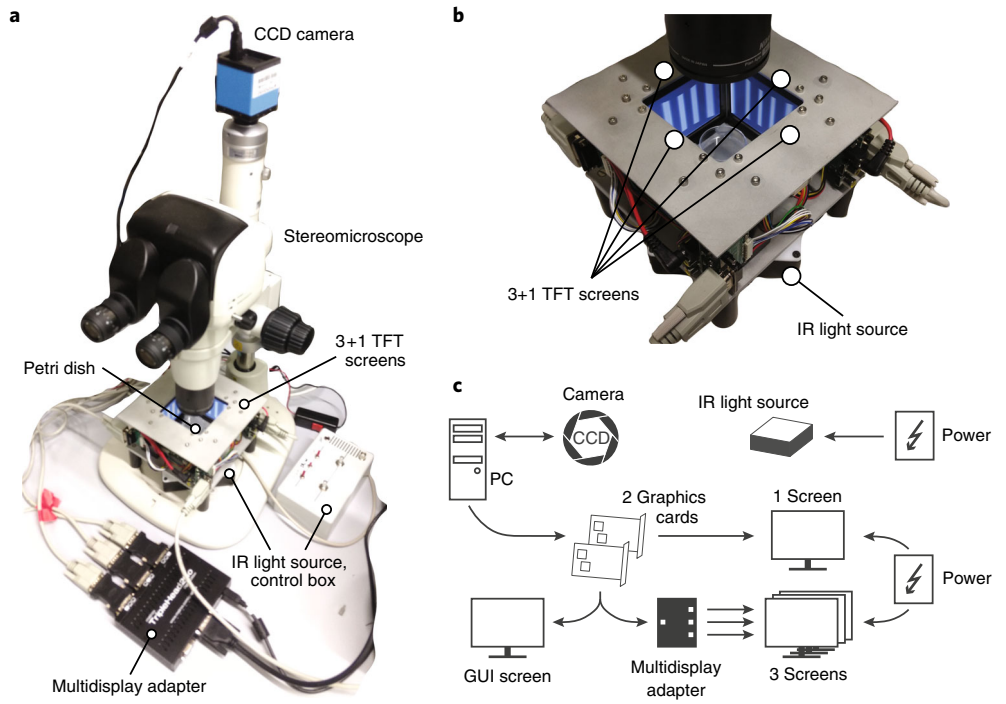

Fig. 3 | Setup for a typical oculomotor experiment. a, Overview of the setup. Zebrafish larvae are placed under a stereomicroscope, in the center of a stimulus arena made up of four computer screens. Psychtoolbox visual stimuli are presented on these screens, and the fish are illuminated from below with infrared (IR) light via a custom control box; this light, in turn, is detected by a CCD camera mounted on the microscope and connected to a computer. **b**, Close-up view of the stimulus arena. The Petri dish containing the fish is placed on an IR-transparent pedestal stage immediately below the microscope objective. **c**, Connectivity. Two graphics cards in conjunction with a multidisplay adapter are used to display the 360° surround stimulus on the four screens.

direction, i.e., perpendicular to the body axis. Positive or negative angles indicate that the eye is turned further to the left or to the right of the fish, respectively (assuming the camera is positioned dorsal to the animal). For instance, if the right eye is oriented nasalward, it would be assigned a positive angle (Fig. 5j). Angular orientation in degrees is saved for both eyes of each fish; Cartesian eye position in pixel coordinates is additionally saved for the first fish so users can detect struggle behavior and other motion artifacts post hoc. The resulting tracking data for up to six simultaneously recorded larvae are combined into a single output file, along with the time stamp of each frame, the stimulus phase that was being presented at this time, and other variables (Table 1).

Low image contrast, fish position near the edge of illumination, and dirt or other stray objects can interfere with the particle-detection algorithm. As described in the main Procedure, users should go through several simple steps to ensure proper eye detection (Steps 14–19). While tracking is under way, yet before starting the recording, users can refer to the 'Debugging tools' tab to verify eye detection. Systematic benchmarking confirms that, if this procedure is followed, eye detection and tracking are very reliable under a wide range of conditions (Fig. 6). We evaluated eye-tracking performance by individually recording anesthetized larvae 5 d post fertilization (d.p.f.) and averaging the standard deviation of detected eye position over 2 min each, as a function of infrared illumination intensity, optical magnification, user-defined detection threshold on pixel brightness, and vertical distance from the focal plane. Depending on modality, either four or five fish were tested. The angular error is typically ~0.1° with optimal settings in our setup.

Offline video analysis

The same algorithm used to track eyes during an experiment can also be used to detect and track eye movements in existing video files of behaving zebrafish larvae (Box 1). Tracking results are again saved as a text file. The present software was developed for use with non-compressed 8-bit .avi files, but multiple types of compressed .avi files can be imported. The following four video codecs are included: Motion JPEG, FF Video Codec 1, Y800 Uncompressed, and YUV 4:2:0 Planar. Virtually,

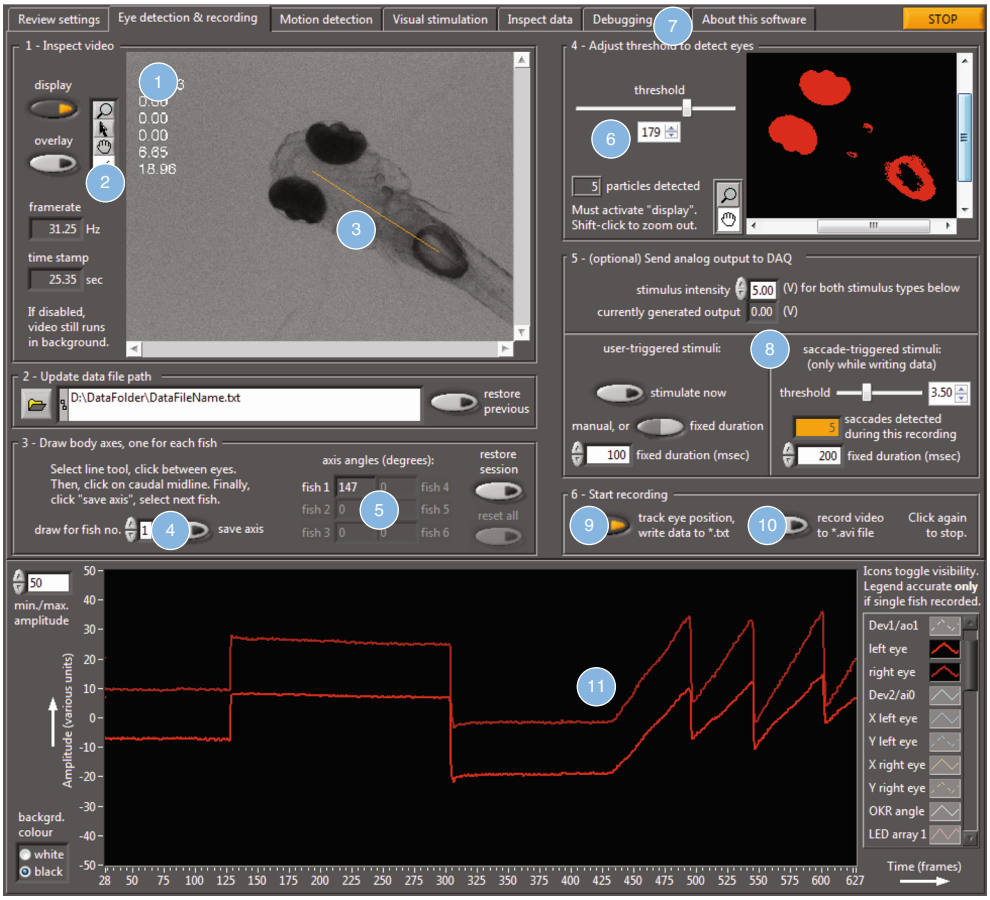

Fig. 4 | Eye detection and recording. From the main user interface, users can (1) display the live or recorded video; (2) use the line tool to (3) draw and (4) save body orientation for (5) each one of up to six fish; (6) adjust the threshold for the eye-detection algorithm; (7) (if needed) use debugging tools to confirm proper eye detection; (8) trigger analog output, e.g., for optogenetic stimulation; (9) start and stop eye tracking, and automatically save data to a new .txt file; (10) start and stop recording the video to an .avi file; and (11) inspect live eye traces while data are acquired. The head of the larva, shown here at 5 d.p.f., is approximately 600 μm wide. Appropriate regulatory board permission was obtained before zebrafish experiments.

any other video format can be converted into one of these using free software (such as Any-VideoConverter, Table 2, no. 3), and can be analyzed in ZebEyeTrack, as long as the frame rate is not altered during conversion. Users can manually install further codecs into their system as usual; as LabVIEW uses the Video Compression Manager of Microsoft Windows, these codecs will then become accessible to ZebEyeTrack. For those users who wish to analyze only existing videos, and who do not require the full ZebEyeTrack software to present and modulate stimuli, control data acquisition, or trigger optogenetic stimulation, we recommend using the reduced, stand-alone version ZebEyeTrack Light instead.

Visual stimulation

ZebEyeTrack supports two different ways to generate stimulus protocols: The first option is a customizable sequence of stimulus phases, each with user-defined stimulus parameters, that is executed from beginning to end (Step 25B or D). Alternatively, stimuli can be generated with parameters that are set ad hoc (Step 25A or C). Both of these stimulus types display horizontally moving vertical bars. And both of these stimulus types can be presented using either one of two hardware setups: a custom array of LEDs well-suited to two-photon scanning microscope experiments (and thus called 'line scan'), or computer screens receiving input via the MATLAB Psychtoolbox. If desired, computer screens can be replaced by projection onto translucent experimental stimulus screens (e.g., paper drums), using standard video projectors with ZebEyeTrack's built-in stimulus-warping options, explained below. Using the custom LED array (Step 25A or B), stimulus velocity and duration can be varied, but contrast and spatial frequency are kept constant (12 cycles/360°; i.e., 0.033 cycles/degree).

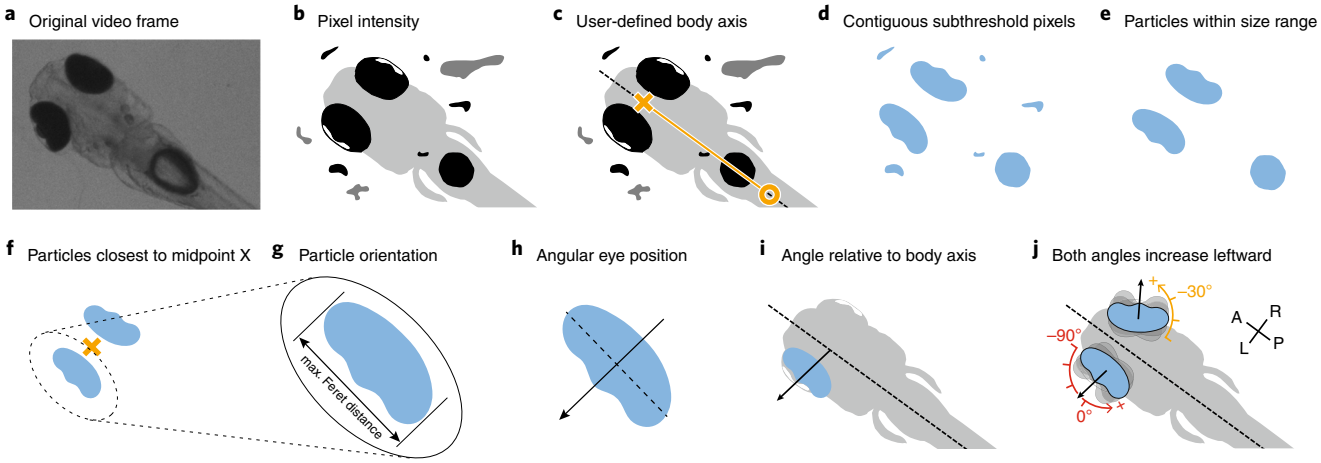

Fig. 5 | Eyes are tracked using a custom particle-detection algorithm. a, Raw video frames are acquired. **b**, Each frame is analyzed based on pixel intensities. **c**, Before eye tracking starts, the user manually defines the body axis of the fish by first clicking on the midpoint between the eyes, X, and then clicking on a caudal part of the midline, O. This axis is used for all frames, or until a new axis is defined. **d**, On the basis of a user-defined threshold, the darkest pixels are identified. **e**, These are combined into particles within a plausible size range. **f**, Only the two particles closest to X are kept. **g**,**h**, Angular eye position is defined as orthogonal to the maximal Feret distance of each particle. **i**,**j**, Angular eye position is defined relative to the body axis, and eye positions of both eyes increase toward the left if the animal is recorded from above. Thus, left-eye position increases during temporalward movements, and right-eye position increases during nasalward movements. Both are zero if orthogonal to the body axis. Note that the fish in **j** is shown from above; in many setups, the fish will instead be filmed from below. A, anterior; L, left; P, posterior; R, right. The head of the larva, shown here at 5 d.p.f., is approximately 600 μm wide. Appropriate regulatory board permission was obtained before zebrafish experiments.

These stimuli can be presented continuously or in combination with saccade-triggered pauses and reversals (Step 25A(iii) and B(ix)). Using computer screens or video projectors instead (Step 25C or D), velocity, duration, contrast, spatial frequency, and other parameters can be adjusted. For all types of stimulation, users can import existing stimulus sequences from a .txt file, or design new sequences through the GUI.

Visual stimulation using computer displays

When using four computer screens or two projection screens for stimulus display, ZebEyeTrack configures the stimulus using MATLAB in conjunction with its freeware, Psychtoolbox. Any stimulus protocol may contain an arbitrary number of stimulus phases, each with different parameters. The available parameters are described among the procedures below (Step 25C or D). There is no technical limit on the length of individual phases or the stimulus protocol as a whole, but for practical reasons, most larvae will be stimulated and recorded only for up to 3 h. After this time, special precautions must be taken to prevent the evaporation of the mounting medium from causing a focus shift of the camera. When presenting horizontally moving vertical bars on screens arranged in a square along the horizon of the animal, the y position of a pixel on the screen does not correspond directly to its perceived elevation angle, as parts of the flat screen are further away from the fish than the screen center and hence appear less elevated. To compensate for this, the outline of the stimulus is cropped to a shape with edges at a constant elevation when seen from the larva's position (Fig. 3b). In addition, as all bars should have an equal and constant angular width from the perspective of the fish, we corrected their shape by making bars wider near the edges of the square stimulus arena. The same adjustments apply to flat projection screens; different adjustments are applied for cylindrical projection screens. Video output is generated by two graphics cards and a multidisplay adapter (Fig. 3c).

Visual stimulation using a custom LED array

The purpose of the custom LED array (Equipment setup) is to enable concurrent visual stimulation and calcium imaging in the absence of stimulation noise during detection of fluorescence signals with the photomultiplier tubes of the microscope. This is achieved by temporally interleaving the line scans of the microscope with visual stimulation[31] and necessitates an LED array that can be rapidly switched on and off (as opposed to regular computer screens or video projectors that cannot be toggled as quickly). In a typical configuration, we record a calcium image time series of 512 × 512 pixels at 2 f.p.s. After each line scan, the scanning mirrors must change direction for the next line, so the signal

Table 1 | Content of the data file if a single fish is tracked

Col.	–	Pa	Pp	La	Lp	Data content	Units
1	•	•	•	•	•	Time stamp (using computer system clock, not starting at zero)	Seconds
2–4	•	•	•	•	•	Analog stimulus channels	Volts
5–6	•	•	•	•	•	Angular eye positions (a_{LE}, a_{RE})	Degrees
7	•	•	•	•	•	Digitized analog input channel	Volts
8–11	•	•	•	•	•	Cartesian center-of-mass eye positions (x_{LE}, y_{LE}, x_{RE}, and y_{RE})	Pixels
12		•	•			Stimulus running	Boolean yes (1) or no (0)
13–16		•	•			Stimulus screens 1–4 active	Boolean yes (1) or no (0)
17		•	•			Stimulus direction reversed	Boolean yes (1) or no (0)
18		•	•			Stimulus angular position	Degrees
19				•	•	Stimulus LED array 1 active	Boolean yes (1) or no (0)
20				•	•	Stimulus LED array 2 active	Boolean yes (1) or no (0)
21				•		Stimulus direction switched	Boolean yes (1) or no (0)
22				•		Stimulus conjugate or disconjugate	Boolean conjugate (1) or not (0)
23				•	•	Current LED stimulus frame (decimal conversion of the 8-bit digital signal sent to the LED arrays)	Integer
24				•		Stimulus temporal frequency	Cycles/s
25		•	•	•	•	Phase (i.e., row) of the stimulus protocol	Integer

All data are exported to a single, tab-delimited .txt file. File names are automatically appended by a time stamp corresponding to the system clock at the time of analysis, so no files are accidentally overwritten. Each row of the file represents one time step, and the 25 columns each represent a different variable. If multiple fish are tracked, the number of columns can increase up to 35 (Supplementary Table 1). Depending on stimulus type, only some columns are written (•); unused columns are either padded with zeros or are simply noninformative. Symbols indicate stimulus type: –, no stimulus; Pa, Psychtoolbox ad hoc stimulus; Pp, Psychtoolbox stimulus protocol; La, line scan ad hoc stimulus; Lp, line scan stimulus protocol. Data include the type of visual stimuli presented, analog input such as the mirror position of a scanning microscope, tracked eye positions and orientations, and analog output to other devices such as event-controlled optogenetic stimulation. *a*, angle; col., column; LE, left eye; RE, right eye; *x*, horizontal pixel coordinate; *y*, vertical pixel coordinate.

from the photomultiplier tubes cannot be used for about 160 µs. The visual stimulus is presented only during these 160 µs of flyback or turning, at a refresh rate of 1,024 Hz. This frequency lies far above the maximal flicker fusion frequency (~20 Hz) in larval zebrafish[32] and is therefore perceived as continuous. To obtain interleaved stimulation, we combine the microscope DAQ signal (a digital channel that is at 0 V during flyback or turning of the scan mirrors and at 5 V during the line scan) with the information from our software (ZebEyeTrack determines which LED bars should be on at any given time) using logic gates[17].

An interleaved stimulus protocol then consists of an arbitrary number of stimulus phases: time intervals of variable length during which the stimulus type (direction, speed, and visibility ON/OFF) is kept constant. To enable disparate stimuli for the left and the right eye, the direction and visibility of the stimulus can be controlled individually for each half of the LED array (i.e., for two of the four LED panels at a time). These stimulus phases can be strung together in sequence and will often include pauses (i.e., intervals with zero speed) to separate different stimuli. The parameters defining each of the interleaved stimulus phases are described among the procedures below (Step 25A or B).

Closed-loop stimulus presentation and analog output

By default, all visual stimuli are preprogrammed and then presented. However, users may wish to adapt stimulation to animal behavior in real time. Examples include eye-position-dependent gain modulation of the moving bars, or saccade-triggered optogenetic stimulation. ZebEyeTrack offers two closed-loop modes. One helps to dissociate motor-related variables (i.e., postsaccadic time and eye drift during the OKR) during imaging experiments in which temporary freezing of stimulus motion is triggered after each saccade (Fig. 2c). This is particularly important in calcium imaging experiments (see the 'Combination with calcium imaging' section below), in which the slow time constant of calcium indicators often conflates otherwise distinct patterns of neural activity. Using ZebEyeTrack, saccade-related motor activity in the brain can be temporally separated from activity related to slow eye movements or eye position (Step 28A(ii) or B(i)). The second mode triggers external events after each saccade by sending an analog output signal via the DAQ box (Fig. 2d). We have used this

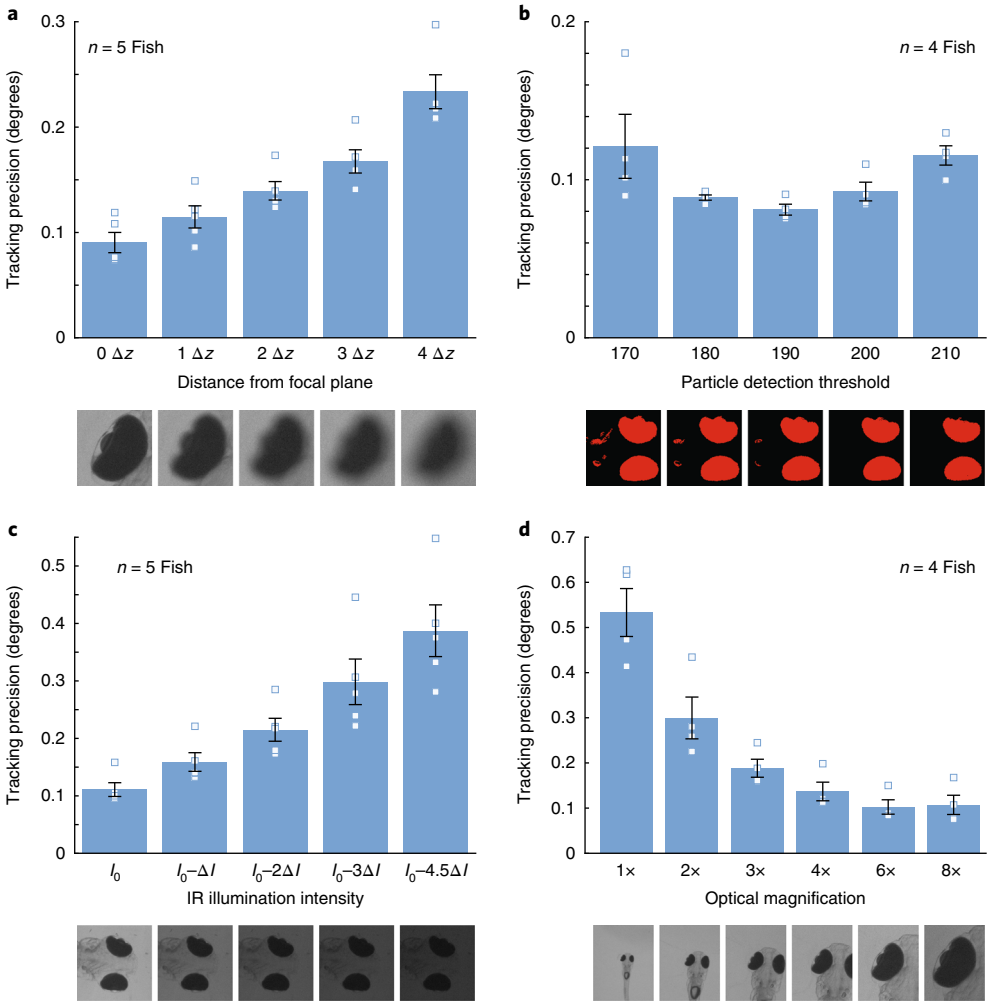

Fig. 6 | Benchmarking. *mitfa*$^{-/-}$ zebrafish larvae (5 d.p.f.) were anesthetized using tricaine, and their immobile eyes were tracked using ZebEyeTrack. The standard deviation of the detected eye position, averaged across both eyes and over time, provides an estimate of eye-tracking precision. **a**, Tracking precision is very robust, even if the fish is located outside the focal plane. Fish were initially perfectly focused, then increasingly defocused. Defocusing increments, i.e., incremental *z*-shifts of the focal plane while the fish remained in place, were kept constant. Five fish were recorded. Solid blue bars indicate the mean, and error bars represent the standard error of the mean. Light blue squares indicate the precision estimate for each individual fish. **b**, With bright IR illumination, eyes are precisely tracked, even if the particle-detection threshold is not precisely tuned. White-light illumination improves precision (decreases detection noise) further by a factor of two (not shown). Four fish were recorded. **c**, Tracking precision improves with more intense IR illumination. Five fish were recorded. ΔI represents changes in the light source potentiometer setting. **d**, The precision of eye tracking depends on optical magnification. As long as the image is focused, higher zoom levels correspond to a larger number of pixels per eye, improving precision. Four fish were recorded. Note that dark particles near the eye can seriously affect the accuracy of angular eye position detection. They have not been quantified here but may result from skin pigments in wild-type animals, air bubbles in the mounting medium, or condensed water on the Petri dish surface. The head of the larva, shown here at 5 d.p.f., is approximately 600 μm wide. Appropriate regulatory board permission was obtained before zebrafish experiments.

functionality to photostimulate a zebrafish 1 s after the saccade for a period of 6 s. As a control, analog output is generated for only two out of every three saccades, so that non-stimulated behavior can be measured after the third saccade (Steps 20 and 21). For the remaining one out of three saccades, negative output values are written to the data file to mark saccade times.

Combination with calcium imaging

We have successfully used our software in calcium imaging experiments, addressing two common issues. First, closed-loop feedback helps distinguish calcium signals related to either saccades or slow-

Table 2 | Software and licensing websites

No.	Description	URL
1	VisioBox (ViewPoint)	http://www.viewpoint.fr/en/p/software/visiobox
2	VisioTracker (TSE Systems)	https://www.tse-systems.com/product-details/fish-behavior-systems/visiotracker/?open=3785#visiotracker-3785
3	AnyVideoConverter (AVC)	http://www.any-video-converter.com/products/for_video_free/
4	BY-NC-SA license 4.0 (Creative Commons)	https://creativecommons.org/licenses/by-nc-sa/4.0/
5	ZebEyeTrack download, updates and support	http://www.zebeyetrack.com/
6	LabVIEW download (National Instruments)	http://www.ni.com/download-labview/
7	MATLAB download (MathWorks)	https://de.mathworks.com/downloads/
8	LabVIEW Run-Time Engine 2014 SP1, 32-bit (National Instruments)	http://www.ni.com/download/labview-run-time-engine-2014-sp1/5198/en/
9	Vision Development Module Run-Time Engine 2015 SP1 (National Instruments)	http://www.ni.com/download/vision-development-module-run-time-2015-sp1/5973/en/
10	Vision Development Module Run-Time Engine 2012 (National Instruments)	http://www.ni.com/download/vision-development-module-run-time-2012/2980/en/
11	Vision Acquisition Software 2015 f1 (National Instruments)	http://www.ni.com/download/ni-vision-acquisition-software-august-2015-f1/5486/en/
12	Software evaluation, latest versions (National Instruments)	https://lumen.ni.com/nicif/d/evaltlktsigproc/content.xhtml
13	MATLAB trial request (MathWorks)	https://www.mathworks.com/programs/trials/trial_request.html
14	DAQmx drivers (National Instruments)	http://www.ni.com/download/ni-daqmx-14.5/5212/en/
15	.NET Framework (Microsoft)	https://www.microsoft.com/net/download/framework
16	VirtualBox download (Oracle)	https://www.virtualbox.org/wiki/Downloads
17	Psychtoolbox for MATLAB	http://psychtoolbox.org/
18	IC Capture (The Imaging Source Europe)	https://www.theimagingsource.com/support/downloads-for-windows/end-user-software/iccapture/
19	IC LabVIEW Extension (The Imaging Source Europe)	https://www.theimagingsource.com/support/downloads-for-windows/extensions/icextlvi/
20	NI-IMAQ for USB Cameras (National Instruments)	http://www.ni.com/example/30030/en/
21	Fiji download (The ImageJ Project)	https://imagej.net/Fiji/Downloads

Links to third-party software, evaluation versions, licensing websites, and online tutorials are listed in the order they appear in this article. All websites were last accessed on 8 February 2018.

phase motion (see the previous section). Second, many commercially available microscopes come with only limited options for triggering and controlling recordings from outside of the proprietary microscope software. In calcium imaging experiments with concurrent visual stimulation and behavioral recording, it is, however, critical to know when the calcium signals occurred relative to the timing of the visual stimulation or the behavior. We implement a simple solution to this problem by synchronizing the data post hoc. To achieve this, we use a Y-split BNC cable on the signal for the (slow) scanning mirror (typically, the vertically scanning mirror) and digitize this signal in our DAQ box (analog input channel 0). The zigzag voltage pattern resulting from image scanning is then saved with time stamps to the .txt data file. After the recording is finished, one can use custom algorithms to detect the peaks and troughs to identify the eye-tracking data points that correspond to each frame of the image time series (Supplementary Fig. 2b). We have used this method to, e.g., generate calcium tuning curves for the temporal frequency of the stimulus[17].

Materials

Reagents

- Zebrafish (wild-type, *mitfa*$^{-/-}$). All data shown in this article were acquired from transgenic fish (*Tg(elavl3:GCaMP5G)a4598Tg*, available from the authors upon request). We chose this line because it is commonly used for experiments in our laboratory and thus is representative of the performance of ZebEyeTrack in practice.

! CAUTION Any experiments involving live zebrafish must conform to relevant institutional and legal regulations. Animal experiments for testing and benchmarking ZebEyeTrack were licensed by the local authorities (Regierungspräsidium Tübingen) in accordance with German federal law and Baden-Württemberg state law. For further details on how to prepare and handle fish, see previously published methods[20].

- NaCl (AppliChem, cat. no. A3597)
- KCl (Roth, cat. no. 6781.1)
- $CaCl_2$ (AppliChem, cat. no. A1873)
- $MgSO_4$ (Merck, cat. no. 1.05886.0500)
- Methylene blue (AppliChem, cat. no. A4084)

! CAUTION Methylene blue is harmful. Use personal protective equipment.

- Low-melting-temperature agarose (Biozym Sieve GeneticPure Agarose; Biozym, cat. no. 850080)

Equipment

Hardware

- To analyze existing videos using ZebEyeTrack or ZebEyeTrack Light, no hardware other than the computer itself is needed. Using a 64-bit Windows 7 Professional operating system, Intel Core i5-4690 3.50-GHz CPU, and 8 GB RAM, we obtain frame rates of about 30 f.p.s.

Free software

- ZebEyeTrack (the software described in this paper). The source code is available for free under a Creative Commons BY-NC-SA license (Table 2, no. 4); users may adapt it to accommodate other modalities, e.g., implement the additional tracking of tail kinematics, and modify or remove any undesired features. The current version of the software is available for download at http://www.zebeyetrack.com (Table 2, no. 5). Through the website, users can also log onto a virtual machine for up to 1 h, test ZebEyeTrack Light, and analyze their own video files.

Commercial software

- National Instruments (NI) LabVIEW 2014 or later (to use ZebEyeTrack, the 32-bit version must be used; Table 2, no. 6), NI Vision Development Module (VDM) 2015 SP1 or later (earlier versions of the VDM additionally require NI Vision Acquisition Software 2015 f1 or later, Table 2, no. 11), and MathWorks MATLAB 2014b or later (Table 2, no. 7). Only the cheaper, basic version of MATLAB is required; none of the commercially available toolboxes need to be purchased. (Optionally, the Image Processing Toolbox (https://www.mathworks.com/products/image.html) can be acquired to enable an additional debugging feature; see Step 19.) If only the compiled version, ZebEyeTrack Light, is used, MATLAB is still required, but each NI software can be replaced by its corresponding run-time engine (RTE). These are available either free of charge (NI LabVIEW RTE, Table 2, no. 8) or at low cost (MATLAB, NI Vision Development Module RTE, Table 2, no. 9). Free trial versions of all programs are available (Table 2, nos. 6 and 10–13). For more detail, see Supplementary Manual 2. For NI Vision Development Module RTE version 2012 and older, NI upgrades must be performed before use. Note that evaluating more recent versions of this specific RTE requires requesting a temporary activation key from NI customer service, so we recommend using version 2012 for evaluation instead. Evaluation versions of all RTEs generally require downloading the full version first, and then indicating during installation that users actually wish to evaluate the software. Choosing the explicit evaluation links on the website instead leads to remote demo machine access instead and is not recommended. LabVIEW or its RTE must, under all circumstances, be installed before all other NI software or drivers.
- *NI DAQmx drivers*. Available for free download (Table 2, no. 14). When installing, we recommend using the browser download instead of the NI download manager option. If the download manager is used, the latest version of the Microsoft .NET Framework (Table 2, no. 15) must be installed beforehand. NI DAQmx also includes the stand-alone application National Instruments Measurement & Automation Explorer, NI MAX (see below), to control and configure DAQ devices.

Optional equipment

- *Virtual machine software*. To run the software under operating systems other than Microsoft Windows, a freeware virtual machine software such as *VirtualBox* by Oracle (Table 2, no. 16) and a license to run Windows on the virtual machine are required. For more detail, see Supplementary Manual 3.

- *Video-conversion software*. To convert existing videos to .avi files with a supported codec, any free conversion software such as *AnyVideoConverter* (Table 2, no. 3) can be used. The following four codecs are supported by default: Motion JPEG, FF Video Codec 1, Y800 Uncompressed, and YUV 4:2:0 Planar.
- *DAQ devices*. One to two DAQ devices, depending on the number of applications chosen (see below), are needed; e.g., device NI USB-6008 by National Instruments.
- *Psychtoolbox*. Presentation of visual stimuli with ZebEyeTrack on computer or projection screens additionally requires the freeware Psychtoolbox for MATLAB[33–35] (Table 2, no. 17), including the open-source GStreamer framework.
- *Graphics cards*. To provide visual stimulation with ZebEyeTrack on computer screens, graphics cards and four attached computer screens (in addition to the one displaying the ZebEyeTrack user interface) connected via display adapters are required, e.g., two graphics cards (NVIDIA, model no. GeForce GT 620) and four thin-film-transistor (TFT) liquid-crystal display (LCD) color screens (Accele Electronics, model no. LCD35VGAN), connected using one multidisplay adapter (Matrox, model no. TripleHead2Go T2G-D3D-IF) are needed.
- *Video projectors*. To provide visual stimulation via projection, one or up to four standard video projectors, and the same graphics cards as above are needed. No additional graphics cards are required, as the video projectors can be connected using the same ports otherwise used for the display screens. When using only one or two video projectors in lieu of computer screens, the multidisplay adapter is unnecessary.
- *LED arrays*. To provide visual stimulation with ZebEyeTrack or ZebEyeTrack Light via LED, custom arrays connected via a DAQ device are needed; e.g., four custom-made LED arrays (flat, rectangular, 43 × 67 mm) made up of 12 columns of seven red LEDs each (LED Superbright; Kingbright, cat. no. 3MM L-934SRD-G), emitting light at 660 nm. In addition, a custom eletronic circuit using logical gates and transistors to gate LED current based on the ZebEyeTrack DAQ signal and the microscope line scan signal are needed. This custom circuit design is available from the authors upon request.
- *Camera*. To record eye movement with ZebEyeTrack or ZebEyeTrack Light, a DirectShow-compatible camera (e.g., The Imaging Source, model nos. DMK 21AU04, DMK 21BF04.H, or DMK 23UV024) is needed. Most cameras are compatible, but we recommend using one of these and the full version of ZebEyeTrack for optimal performance. The camera can be mounted on a stereoscope with infrared illumination from below[20] or can be used with a macro lens. In addition, to control the CCD camera, the freeware IC Capture video acquisition software (Table 2, no. 18) and its LabVIEW Extension (Table 2, no. 19) by The Imaging Source are needed.
- *Laser diode and optic fiber*. To provide optogenetic perturbation, a laser diode and optic fiber, controlled via DAQ device, are needed (e.g., fiber-coupled LED (Thorlabs, cat. no. M470F1) and diode power controller (Thorlabs, model no. LEDD1B)).

Reagent setup

E3 buffer solution

E3 buffer solution is 5.0 mM NaCl, 0.17 mM KCl, 0.33 mM $CaCl_2$, and 0.33 mM $MgSO_4$. If desired, add 200 μL of 0.05% (wt/vol) methylene blue in H_20 as fungicide to 1 L of E3. E3 buffer solution can be stored for several weeks at room temperature (25 °C).

Immobilization gel

Prepare 50 mL of 16 g/L (1.6% (wt/vol)) low-melting-temperature agarose in E3 buffer solution. To do so, heat the solution to 80 °C and mildly agitate it by manually swirling the beaker with a circular motion, avoiding the creation of excessive air bubbles. Prepare aliquots of the solution in 1.5-mL tubes and place them in a block incubator at 39 °C. Wait for 5 min to let the solution reach 39 °C before bringing it into contact with the fish. The aliquots can be kept at 39 °C for a week or longer.

Equipment setup

Visual stimulation on computer screens

The first graphics card outputs an HDMI signal containing information for three computer screens (2,400 × 600 pixels; Fig. 3). Via an HDMI to DVI-D adapter, this signal is connected to a Matrox TripleHead2Go multidisplay adapter, which outputs three VGA signals that connect to three individual computer screens. The second graphics card outputs a DVI-A signal (800 × 600 pixels) for the fourth computer screen. For convenience, the GUI for LabVIEW should be run on a fifth display, such as a regular computer screen. In our setup, this is also connected to the first graphics card via

DVI-A. Please refer to Supplementary Manual 1 for details on cable connections and MATLAB screen settings. Other configurations that include alternative hardware are possible but must be troubleshot to make the stimulus appear in the correct and anticipated parts of the available computer screens. As an optional variant, ZebEyeTrack fully supports replacing these computer screens with standard video projectors and either flat or cylindrical projection screens.

Visual stimulation on custom LED array

The LED arena surrounds the animal horizontally (360°, Supplementary Fig. 2). Each screen has four channels that control three LED columns each (channel a: columns 1, 5, and 9; b: 2, 6, and 10; c: 3, 7, and 11; and d: 4, 8, and 12), so that four distinct frames can be displayed for any optic flow stimulus, resulting in a spatial frequency of 12 cycles/360°. LEDs in each column are wired in parallel, and columns of the four individual channels are wired in series. Separators are inserted between LED columns to increase contrast. Luminous emittance of a single array, measured on the surface of the diffuser, was ~11 lux. Irradiance, measured with a power meter at the position of the fish, was ~6 $\mu W/cm^2$. Plastic-covered paper screens placed about 8 mm in front of the LED arrays act as diffusers to make bars appear more homogeneous. The resulting Michelson contrast between active and inactive LED columns, as perceived through the diffuser, is ~0.6. An optional nasal separator can be placed in the central 45° portion of the visual field, disconnecting left and right visual hemi-fields to enable separate stimulation of each eye. The two LED arrays in each visual hemi-field are coupled and display identical stimuli, so that eight digital channels are sufficient to control the LED arena. Please note that the available current of the digital channels is too low to drive the LEDs directly, and therefore a transistor logic circuit must be placed between the DAQ digital out channels and the LED matrices. The circuit uses logic gates to generate a 5-V signal whenever a given column of LEDs should be ON, i.e., at all times when both of the following statements are true: (i) the scanning mirror is turning and (ii) the LED column is ON in the current LED image frame (defined by eight digital output pins of the DAQ box controlled by ZebEyeTrack). This signal is then used to gate the channel current to the LEDs via a transistor. A detailed description of the LED arena and the circuit is not part of this protocol but is available from the authors upon request. For optic flow stimulation, each LED array can generate four different images, each with one channel on and three channels off, resulting in a very low frame rate (e.g., 16 f.p.s. for a temporal frequency of 4 cycles/s). However, animals readily perform OKR in response to this suboptimal stimulus.

Post hoc hardware synchronization

To later correlate eye movements with other optional processes, such as the scanning motion of a two-photon microscope, acquire the appropriate analog input via a DAQ device. ZebEyeTrack will save it to the same file as the eye-tracking data. Make sure the name of your DAQ device, as listed under 'Devices and Interfaces' in NI MAX, matches the DAQ device name expected by ZebEyeTrack ('Dev2'). The signal is read in from the first analog input channel, Dev2/ai0.

Camera support

ZebEyeTrack already supports DirectShow-compatible cameras by numerous manufacturers, often with automatic plug-and-play configuration. This includes multiple recommended cameras by The Imaging Source. Experienced users wishing to incorporate further camera-specific sub-VIs provided by their preferred manufacturers may choose to adapt the source code of ZebEyeTrack, but this is purely optional. In such a case, sub-VIs to adapt or replace are located within the DetectCamera.vi, CameraInitialisation.vi, and GrabImage.vi files. Supporting software and documentation to acquire images from a variety of cameras via DirectShow is available for download from National Instruments (Table 2, no. 20).

Operating systems

Running ZebEyeTrack on operating systems other than Windows, such as Mac OS or various Linux distributions, is possible and merely requires setting up a virtual machine, which in turn runs a licensed copy of Microsoft Windows as its operating system. This procedure is straightforward and can be completed within 1 h. Virtual machine licenses are available for free (Supplementary Manual 3). Performance is limited only by the fraction of host computer resources allocated to the virtual machine; if permitted, ZebEyeTrack will perform as well as on a regular Windows PC.

Procedure

Software installation and hardware setup ● Timing variable

1 Download to your hard drive and unpack the compressed archive containing ZebEyeTrack or ZebEyeTrack Light. The current version of the software is available through www.zebeyetrack.com.
▲CRITICAL STEP If running anti-virus software, add ZebEyeTrack Light to its list of trusted programs to ensure full hardware access.

2 Check which additional software is required for your specific application (Supplementary Table 2) and install it (see Equipment and Equipment setup). Install all drivers required for your hardware. For ZebEyeTrack Light, see instructions in Supplementary Manual 2.

3 Take the following steps to ensure ZebEyeTrack recognizes your hardware: Close LabVIEW and open IC Capture to check whether the camera is properly detected. Open the NI Measurement and Automation Explorer (NI MAX) and self-test the DAQ device. When using the reduced version ZebEyeTrack Light, and only analyzing existing videos, no hardware is required.

4 (Optional) If you intend to use DAQ devices to generate or acquire analog signals, check whether the devices are named either 'Dev1' or 'Dev2'. Before starting ZebEyeTrack, open NI MAX, find connected devices under 'My System/Devices and Interfaces', verify these names, or rename your device after double-clicking its name.

Immobilizing zebrafish larvae ● Timing 5–10 min

5 Immobilize zebrafish larvae with agarose gel in a Petri dish by following any of the common procedures already published[20]. Briefly, prepare liquid low-melting-temperature agarose in E3 medium (1.6% (wt/vol) in 1-mL vials) and place it inside an incubator set to 39 °C. Keep it inside the incubator for approximately 5 min, until it has cooled to 39°, so as not to harm the fish in the next step by exposing it to excessive heat. For future experiments, the solution can be stored in the incubator for several days. Next, pipette a fish into the vial and then place a drop of agarose (including the fish) onto the surface of a new Petri dish. Ensure upright position of the fish with a manipulation tool (e.g., platinum wire tool), let the agarose harden for 5 min, add E3 medium to the dish, and then remove the agarose surrounding the eyes using a second platinum wire tool, which is flattened at the tip for agarose incisions. Place the immobilized larvae in the arena. Confirm that fish are well illuminated and within the field of view of the camera.
▲CRITICAL STEP Agarose temperature will drop after removal of the fish from the agarose vial (incubated at 39 °C). Only 10–20 s are available to orient the fish. Make sure to finish this step before the gel starts to congeal. Imperfect illumination can interfere with eye detection, as the two eye particles might end up having different sizes, inaccurate angles, or shapes. We recommend using minimally pigmented *mitfa*$^{-/-}$ animals where possible to facilitate eye detection, but fully pigmented wild-type animals work as well, if settings are carefully adjusted.

Software configuration ● Timing 1 min

▲CRITICAL Steps 6–13 describe the procedures for software configuration in order to acquire live videos. To analyze existing video files instead, follow the optional procedure in Box 1 instead.

6 To execute VI, click on the white arrow button in the upper left of the LabVIEW window. When first opening the program, it will be executed automatically.

Table 3 | File names

File type	File name
Main data file	filename.txt
Protocol-only data files	filename_timestamp_StimulusProtocol.txt
Protocol spreadsheet	filename_timestamp_StimulusProtocolParams.txt
Video files	filename_timestamp.avi

In Step 7 or 13 of the main Procedure, users choose a file name, "filename", which is then used as the root name for all files created and saved by ZebEyeTrack. Video and protocol-only data file names are automatically appended with the time stamp of their beginning, to create a separate file for each recording. Thus, existing data are never overwritten, and users do not have to manually update "filename" for subsequent recordings.

▲ CRITICAL STEP If buttons on the settings tab are grayed out, the software is already running in a particular operating mode. In this case, click on 'STOP' to abort, and then click the white arrow again to restart.

? TROUBLESHOOTING

7 Select 'Path and *.txt file name' using the free-text field. This file name will be the root name for files created and saved by ZebEyeTrack (Table 3).

▲ CRITICAL STEP Choose a file name with the appropriate file-type extension, .txt. Otherwise, the file name may be truncated.

▲ CRITICAL STEP The file name can be changed between recordings. If the filename is not changed between recordings, new data will be appended to the bottom of the existing main data file. Video and protocol-only data file names are automatically appended with the time stamp of their beginning, to create a separate file for each recording. Thus, existing data are never overwritten, and users do not have to manually update the file name for subsequent recordings.

8 To acquire live video, deactivate 'analyse existing .avi file', and select a camera frame rate from the drop-down menu. 15–20 f.p.s. will be sufficient for most applications. Verify that the correct frame rate is displayed in the adjacent numerical display.

9 Select appropriate settings for your application. To present visual stimuli on custom LED panels, follow option A. To present visual stimuli on computer or projection screens, follow option B.

(A) Presenting visual stimuli on custom LED panels

(i) Activate 'use custom LED arrays'.

(ii) Select the digital output port of the DAQ device.

(iii) If currently active, deactivate 'use computer screens'.

(B) Presenting visual stimuli on computer or projection screens

(i) Activate 'use Psychtoolbox'.

(ii) Move the slider to select either 'ad hoc stimuli' or 'stimulus protocols'.

(ii) (Optional) Differently shaped screens, as well as video projection, require geometric corrections. Click on 'advanced settings' to set screen geometry, dimensions, and projector distance. Close the pop-up window in order to confirm the selection.

(iv) If currently active, deactivate 'use custom LED panels'.

10 (Optional) For mere eye tracking and visual stimulation via computer screens, this step is not needed. To drive external hardware such as that used for optogenetics, and to record the status of external hardware such as a scanning microscope mirror position, ZebEyeTrack uses a DAQ device to generate analog output or acquire analog input, respectively. In these cases, select whether to use DAQ device 'Dev1', DAQ device 'Dev2', or both. Select one of three possible output channels ('Dev1/ao1', "Dev2/ao1', or 'Dev2/ao0'). By default, the analog output is triggered by the occurrence of saccades. Activate 'enable manually triggered output' to be able to manually trigger output signals. Furthermore, disable saccade triggers by clicking on 'disable saccade detection' on the 'Review settings' tab (to use manual and saccade triggers at the same time, leave saccade detection enabled). Enable 'analog input'.

11 (Optional) If you wish to improve recording quality, reduce video file size or increase sampling rate, adjust video settings accordingly. To switch camera modes, such as camera resolution, click on 'advanced settings' next to the camera frame rate selector. Confirm that 'camera detected' displays the correct device name.

▲ CRITICAL STEP If multiple cameras are connected, the order of connection determines which camera is listed here. Disconnect and reconnect if needed. Check 'manual' instead of 'automatic', and select the desired camera mode from the drop-down menu. Close the pop-up window to confirm selection.

12 Click on 'accept these settings and run the programme'.

? TROUBLESHOOTING

13 (Optional) At any time between recordings, 'Update data file path' on the 'Eye detection & recording' panel can be modified. Use this functionality if you want to save the previously recorded data in a different file than that used for the new data, e.g., for starting a new trial for the same animal using the same settings. ZebEyeTrack will create a new folder if it does not exist. Click on 'restore previous' to fill in the name of the last file recorded.

Eye detection ● Timing 2 min

14 (Optional) On the 'Eye detection & recording' panel, live video images are displayed. To increase processing speed, disable the 'display' button.
? TROUBLESHOOTING

15 The top right video panel of the 'Eye detection & recording' tab should display the eyes of each fish as solid red areas surrounded by a black background. Other objects may appear as well and should not pose a problem. Adjust the 'threshold' value if the eyes are not visible.
▲ CRITICAL STEP Inappropriate thresholds may cause less precise tracking or tracking of unrelated structures such as the swim bladder or the edge of the illumination halo.
▲ CRITICAL STEP Loss of focus due to evaporation, extreme eye positions during OKR, or anything else that affects image quality may perturb tracking over the course of an experiment. Consider including a safety margin when adjusting the threshold; i.e., find a range of acceptable threshold values and then pick a value from the middle of this range.

16 (Optional) To recover all body axes used for the previous recording, click on 'restore session'.

17 On the 'Eye detection & recording' panel, set the 'fish no.' to 1. Use the 'line' tool of the video display to manually indicate the orientation of the first fish. Draw a line parallel to the body axis by first clicking at the midpoint between the centers of mass of both eyes and then clicking again in a more caudal location. Repeat the process until you are satisfied, and then click on 'save axis'.
▲ CRITICAL STEP Inverting the orientation of the line will swap the left and right eyes.
▲ CRITICAL STEP Selecting a more rostral position than the midpoint between the eyes (such as the nose itself) may result in detection of a nearby stray object or the edge of illumination, rather than both eyes.
? TROUBLESHOOTING

18 Choose a different 'fish no.' for each following fish (up to 6), draw its body axis, and click on 'save axis'. The 'axis angles' table displays the orientation of all axes currently in memory. To delete them all and reduce the total fish number to 1, click on 'reset all'.

19 (Optional) If you possess the MATLAB Image Processing Toolbox, you can obtain direct visual confirmation of eye detection. (Otherwise, skip this step and carefully adjust the threshold in Step 15 instead.) Navigate to the 'Debugging tools' tab and activate 'debug eye detection'. On the adjacent video panel, the eyes should appear as two particles of approximately the same size, but in different shades of the same color. The lighter one is treated as the 'left' eye; the darker one is treated as the 'right' eye. Adjust the threshold until all eyes display stable colors, then click on 'accept new threshold', and return to the 'Eye detection & recording' tab. If the eyes have been switched, or one of them is not detected, set the body axis once more and try again. The eyes of the first fish are shown in shades of red, whereas the eyes of subsequent fish appear in shades of green, dark blue, yellow, light blue, and gray.
▲ CRITICAL STEP Always confirm proper eye detection before starting a recording.
? TROUBLESHOOTING

(Optional) Analog output (e.g., for optogenetics) ● Timing 1 min

▲ CRITICAL Follow the steps in this section if you wish to control external hardware such as that used for optogenetic stimulation. Otherwise, skip to the following section.

20 Configure analog output by setting its 'stimulus intensity' in volts, as well as the following parameters: for manually triggered output, set a 'fixed duration (msec)' for each instance of analog output and flip the slider to 'fixed duration'. Alternatively, flip the slider to 'manual' to constantly generate output until manually terminated using the 'stimulate now' button. For saccade-triggered output, set the saccade detection 'threshold' (applied to frame-to-frame eye position difference in degrees), as well as the 'fixed duration (msec)' of each saccade-triggered output. By default, stimulations are generated 1,000 ms after saccade detection.

21 Generate analog output via a DAQ device. Click on 'stimulate now' to manually trigger the output. Clicking again will terminate the output if the slider is set to 'manual duration'. Saccade-triggered output will be generated automatically while you record eye-tracking data.
? TROUBLESHOOTING

Eye tracking and recording ● Timing variable

22 To start data acquisition, click on 'track eye position' on the 'Eye detection & recording' tab. Data acquisition will end whenever 'track eye position' is manually deactivated. At the beginning of each

Table 4 | Content of the overview result file generated during post hoc analysis

Row	Data content	Units
1	Phase (i.e., row) of the stimulus protocol	None (integer)
2	Saccade detection threshold	Degrees per second
3	Pause between saccade and slow phase	Seconds
4	Maximum length of slow-phase fit	Seconds
5	Mean slow-phase velocity (left eye)	Degrees per second
6	Mean slow-phase velocity (right eye)	Degrees per second
7	Variance of slow-phase velocity (left eye)	Degrees per second
8	Variance of slow-phase velocity (right eye)	Degrees per second
9	Number of saccades detected (left eye)	None (integer)
10	Number of saccades detected (right eye)	None (integer)

Parameters and general results are saved to one tab-delimited .txt file per fish. File names are automatically appended by a time stamp corresponding to the system clock at the time of analysis, so no files are accidentally overwritten. The ten rows of the file each represent a different variable, whereas each column represents one phase of the stimulus protocol.

Table 5 | Content of detailed result files generated during post hoc analysis

col.	Data content	Units
1	Saccade times relative to onset of stimulus phase (left eye)	Seconds
2	Saccade times relative to onset of stimulus phase (right eye)	Seconds
3	Slow-phase velocity after each saccade (left eye)	Degrees per second
4	Slow-phase velocity after each saccade (right eye)	Degrees per second

Detailed results for each stimulus phase are saved to a separate tab-delimited .txt file, one fish at a time. File names are automatically appended by a time stamp corresponding to the system clock at the time of analysis, so no files are accidentally overwritten. Each row of the file represents one detected saccade, and the four columns each represent a different variable.

recording, a single video frame is saved as a .jpg file for future reference. Existing .txt files are never overwritten: in the case that recordings are acquired using the same file name, the data of the second recording will be appended to the data of the first recording (Tables 1).

? TROUBLESHOOTING

23 (Optional) Compare video images to the real-time eye-tracking readout at the bottom of the screen. Right-click icons on the right and select 'Plot visible' to toggle the visibility of individual graphs. In the case that you notice a mismatch, refer to the Troubleshooting section. For longer recordings (exceeding the length of the bottom panel), follow Steps 31–33 to visualize the complete recording period.

24 ZebEyeTrack automatically writes eye-tracking data to a tab-separated .txt file. Its file name is the one chosen in Step 7 or 13, and its content is explained in Table 1.

(Optional) Stimulus design and presentation ● Timing 5–10 min

▲CRITICAL Follow Steps 25–28 if you wish to present any kind of visual stimulation. Otherwise, such as when measuring spontaneous eye movements in the dark, skip to Step 29.

25 Design or import a visual stimulus or stimulus protocol. For ad hoc stimuli and spreadsheet stimuli protocols using custom LED arrays, use options A and B, respectively. For ad hoc stimuli and spreadsheet stimuli protocols using computer or projection screens, use options C and D, respectively. Skip this step entirely if you do not require visual stimulation.

(A) Ad hoc stimuli using custom LED arrays

(i) Open the 'Visual stimulation' tab and its 'Line scan ad-hoc' subtab, and set the constant temporal frequency in cycles/s. This determines the velocity of the moving bars, as the spatial frequency of the custom LED array cannot be adjusted.

▲CRITICAL STEP To obtain a moving stimulus, choose temporal frequencies of 0.5 cycles/s or more; for stationary stimuli, choose a temporal frequency of 0 cycles/s.

? TROUBLESHOOTING

(ii) To modify the direction of stimulus motion, click on 'switch direction'. Click on 'convergent/divergent' to present temporonasally moving (convergent) or nasotemporally moving (divergent) stimuli in both visual hemi-fields. By default, rotational stimuli are shown instead.

(iii) (Optional) To halt and/or reverse the stimulus when a saccade is detected, activate 'trigger pause'. If you want the stimulus to reverse direction after each pause, also activate 'reverse after pause'. The length of the pause before each reversal is given by 'pause in sec'. To change the saccade detection threshold (applied to frame-to-frame eye position difference in degrees), navigate to the 'saccade-triggered stimuli' section of the 'Eye detection & recording' tab, and update 'threshold'.

▲ **CRITICAL STEP** This function is not available if 'disable saccade triggers' was activated on the 'Review settings' tab.

? TROUBLESHOOTING

(B) Spreadsheet stimulus protocols using custom LED arrays

(i) Open the 'Visual stimulation' tab and its 'Line scan protocol' subtab. To reuse a stimulus protocol, click on 'load protocol' to import it from a spreadsheet file.

▲ **CRITICAL STEP** The .txt file can be edited outside of LabVIEW in a spreadsheet. In this case, make sure to keep the tab-separated format (only the number of rows may be changed).

(ii) Each row of the stimulus protocol table represents one stimulus phase, and phases will be presented one after the other, from top to bottom. For each stimulus phase, set the temporal frequency in cycles/s. This determines the velocity of the moving bars, as the spatial frequency of the custom LED array cannot be adjusted.

▲ **CRITICAL STEP** To obtain a moving stimulus, choose temporal frequencies of 0.5 cycles/s or more; for stationary stimuli, choose a temporal frequency of 0 cycles/s.

? TROUBLESHOOTING

(iii) Indicate the duration of each phase in seconds.

(iv) Choose whether to display the stimulus on both LED arena halves or selectively on just the left or the right pair of displays ('1' means a given pair of displays will be used).

(v) To make stimuli presented to both eyes appear rotational or translational, combine nasotemporal and/or temporonasal motion on these different displays ('1' forces nasotemporal; '0' forces temporonasal motion). The final two parameters indicate stimulus direction and whether binocular stimuli are convergent or divergent. They are computed by the program based on all other parameters and are displayed for information only.

(vi) (Optional) If you wish to change how many stimulus phases are presented, choose a value at 'use only rows 0 to', and click on 'apply this row limit'. If the table contained a different number of active rows, it will now be cropped or expanded.

(vii) (Optional) To modify the duration of all stimulus phases in an existing protocol, simply provide a scaling factor, and click on the 'scale all durations' button.

▲ **CRITICAL STEP** To make sure all stimulus phases appear as intended, use this scaling factor to inspect a shortened version of your overall stimulus protocol before running your experiment.

(viii) (Optional) To manually save the stimulus protocol to a tab-separated spreadsheet file before starting the experiment, click on 'save protocol'. (Note: A copy of the stimulus protocol is also automatically saved once eye tracking and stimulus presentation has begun, i.e., during the experiment.)

(ix) (Optional) To halt and/or reverse the stimulus when a saccade is detected, activate 'trigger pause'. If you want the stimulus to reverse direction after each pause, also activate 'reverse after pause'. The length of the pause before each reversal is given by 'pause in sec'. To change the saccade detection threshold (applied to frame-to-frame eye position difference in degrees), navigate to the 'saccade-triggered stimuli' section of the 'Eye detection & recording' tab, and update 'threshold'.

▲ **CRITICAL STEP** This function will not be available if 'disable saccade triggers' was activated on the 'Review settings' tab.

? TROUBLESHOOTING

(C) Ad hoc stimuli using computer or projection screens

(i) Enter stimulus parameters on the 'Visual stimulation: Psychtoolbox ad-hoc' panel.

(ii) Set the 'duration' of the stimulus in seconds.

▲ **CRITICAL STEP** Each time you change a parameter, click on 'update parameters' to confirm their new values before starting the next stimulus.

▲ CRITICAL STEP To avoid overlapping stimuli and flickering screens, do not click on 'update parameters' while the old stimulus is being presented.

? TROUBLESHOOTING

(iii) Set the 'velocity' in degrees per second. This is the angular velocity of the moving bars and is kept constant during one phase.

(iv) Set the 'no. of spatial cycles (bars)'. This defines the spatial frequency of the pattern to display (bars/360°).

(v) Set the 'refresh rate' to 1, so that the code will try to update the stimulus at the frame rate of the diplays. Otherwise, if the display frame rate is 60 f.p.s. and the refresh rate is set to 10, the stimulus will update at no more than 6 f.p.s. Look-up table (LUT) stimulus 'contrast' can take any numerical value between 0 and 1, with 1 yielding white bars over a black background, and 0 yielding gray for both bars and background, rendering the bars invisible. For proper psychophysical contrast experiments, it is important to measure the emitted display light intensities at different display LUT values and then adapt the ZebEyeTrack source code to generate defined contrasts given the gamma values of the display in use[36].

(vi) Set the number of cycles. This indicates the spatial frequency, i.e., the number of bars displayed.

(D) Spreadsheet stimulus protocols using computer or projection screens

(i) (Optional) If you wish to reuse an existing protocol instead of designing it now, navigate to the 'Visual stimulation: Psychtoolbox protocol' panel. Click on 'load protocol' to import it from a spreadsheet file. Before running the loaded protocol for the first time, its frames must be precomputed once again (Step 25D(iv) below). Note that the .txt file can be edited outside of LabVIEW in a spreadsheet. In this case, make sure to keep the tab-separated format (only the number of rows may be changed).

(ii) Enter stimulus parameters on the 'Visual stimulation: Psychtoolbox protocol' panel, where each row of the table represents one stimulus phase. These phases will be presented one after another, from top to bottom. Parameters here are the same as the eponymous ones used for ad hoc Psychtoolbox stimuli.

(iii) (Optional) If you wish to change how many stimulus phases are presented, choose a value at 'use only row 0 to', and click on 'apply this row limit'. If the table previously contained a different number of active rows, it will now be cropped or expanded.

(iv) All frames of the protocol must be precomputed before presentation. Click on 'calculate protocol' and wait for the orange light to come on. Once the stimulus protocol has been run for the first time, it can be rerun by clicking on 'Run protocol' immediately after without having to calculate it again. Whenever you switch between protocols, you must calculate them again.

(v) (Optional) Each time a stimulus protocol is run, the table will automatically be saved to filename_timestamp _StimulusProtocolParams.txt. To manually save it to a file with a different name, click on 'save protocol'.

26 (Optional) To present visual stimuli in different parts of the visual field, such as monocularly or binocularly, you can switch individual displays on or off. Before or during stimulation, navigate to the respective subtab and choose the desired combination of buttons. For 'Line scan ad-hoc', 'Line scan protocol', or 'Psychtoolbox' stimuli, navigate to the subtab opened in Step 25. Then follow option A, B, or C below.

(A) For 'Line scan ad-hoc' stimuli

(i) Toggle '1st half-arena' and/or '2nd half-arena' individually.

(B) For 'Line scan protocol' stimuli

(i) Define the setting within the spreadsheet (Step 25B(iv)).

(C) For 'Psychtoolbox' stimuli

(i) Toggle 'screen 1', 'screen 2', 'screen 3', and/or 'screen 4' individually.

27 After eye tracking has started (and the 'track eye position' button is active, see Step 22), start stimulus presentation by clicking on 'present stimulus' on the respective subtab. Each run of a stimulus protocol triggers the saving of a filename_timestamp_StimulusProtocolParams.txt file containing the parameters of the protocol, as well as an additional filename_timestamp_StimulusProtocol.txt file containing the eye-tracking data acquired during stimulation. The same data are also appended to the main filename.txt file.

▲ CRITICAL STEP If the recording is started while a stimulus is already being presented, data files will be cropped. Avoid this, if possible.

28 (Optional) It is possible to modify ongoing stimulus presentation on custom LED arrays. To do so, follow option A for ad hoc stimuli and option B for spreadsheet stimulus protocols.

(A) Ad hoc stimuli

(i) Modify stimulus temporal frequency, direction, or convergence (Step 25A(i, ii)) at any time while the stimulus is running.

(ii) Stimulus presentation can be halted, and stimulus direction reversed, at fixed intervals. To do this, select the 'finite number of pauses', after which the stimulus will run uninterrupted. The interval between pauses is given by the value of 'every x sec'. The length of each pause is given by 'pause in sec'. To begin, click on 'start schedule of pauses now' while the stimulus is being presented. If you want the stimulus to reverse direction after each pause, activate 'reverse after pause'.

▲ **CRITICAL STEP** The pause schedule begins immediately, even if the stimulus has not been started yet. Activate the stimulus first, then the pauses.

(B) Spreadsheet stimulus protocols

(i) Stimulus presentation can be halted, and stimulus direction reversed, at fixed intervals (Step 25B(ix)).

(Optional) Recording of a video ● Timing variable

▲ **CRITICAL** Follow Steps 29 and 30 if you wish to record an .avi video of fish behavior with or without concurrent eye tracking. Eye position in these videos can also be tracked again later on (Box 1) with a different set of tracking thresholds and body axes each time, if desired.

29 In the case that the extracted eye traces do not represent a sufficient documentation of animal behavior in the experiment, the captured video data can be saved simultaneously. On the 'Eye detection & recording' tab, click on 'record video' in order to record a video with or without concurrent eye tracking. If eye tracking is indeed ongoing, an additional filename_timestamp_video.txt file will be saved next to the video file, which includes only the part of the tracking data that was acquired while the video was recorded.

▲ **CRITICAL STEP** Recording a video may slow the sampling rate of the eye tracking.

? TROUBLESHOOTING

30 Click on 'record video' again to stop the video recording.

(Optional) Inspection of the latest data file or video ● Timing variable

▲ **CRITICAL** Follow the steps in this section at any time if you wish to inspect the results of the latest recording. This also allows you to inspect data in real time while they are still being acquired. If you do not require inspection, skip to the following section.

31 Select a data file: If no recording is ongoing, enter the path of a data file saved earlier under 'Update data file path' in the 'Eye detection & recording' panel. If you want to inspect the last file saved during this session, do not change the path. If a recording is currently ongoing, you can review the data acquired so far. To do so, do not change the path.

32 To inspect individual traces, navigate to the 'Inspect data' tab and select the 'Inspect eye traces' subtab. Click on 'inspect eye traces' to load the data acquired. For an ongoing recording, click again to update. Activate and deactivate individual plots by clicking their legend on the right-hand side of the screen. Zoom in and out by either setting axis limits using the numerical input fields or clicking on 'toggle automatic scaling' and using the scroll bar.

33 To review the latest .avi file recorded frame by frame, navigate to the 'Inspect video frames' subtab. There, click on 'display last video' and use the scroll bar at the bottom to navigate between frames.

Ending the experiment ● Timing instantaneous

▲ **CRITICAL** As described in the 'Experimental design' section under the heading 'Customization of software features', users can add features to the ZebEyeTrack source code (while it is not being executed) and then access and test them during operation. This does include a more general motion-detection algorithm (Box 2). Once you have used all desired features and wish to end the experiment, follow Step 34.

34 Stop the recording of tracking data by deactivating the 'track eye position' button.

▲ **CRITICAL STEP** To properly record all eye-tracking data during visual stimulation, it is important to start the eye-tracking recording before starting visual stimulation and to end visual stimulation before stopping the eye-tracking recording.

Box 3 | Data analysis via the GUI ● Timing 0–2 minutes per stimulus phase

Procedure

1 To load an existing data file, open the 'Inspect data' tab and its 'Instant post hoc analysis' subtab (Fig. 7). In the upper text field, enter the name of your .txt data file and click on 'load data file'.
2 (Optional) To export the results to a file, activate 'while analysing, save to result file'. Whenever you analyze data (step 7 of this box), the results will be saved to the result file indicated in the bottom text field and appended with a label for each stimulus phase *N* and fish *M*: resultfile_phase*N*_fish*M*.txt (Table 5). An additional result file, resultfile_meta_fish*M*.txt, is created to save analysis parameters (Table 4).
3 Select the analysis parameters. Using the sliders, set a 'saccade threshold' in degrees per second, and set the 'length of fit to slow phase' in seconds and the 'pause' to wait after each saccade before fitting the slow phase, also in seconds. To select parameter values exceeding the slider limits, enter them directly into the numerical free-text field next to the slider; to confirm the new value, click anywhere outside the field you edited.
4 (Optional) Select the stimulus phase to analyze. Stimulus phases contained in the data file are automatically detected by ZebEyeTrack. Select one from the drop-down menu if you want to analyze only a single phase.
5 One pair of eyes can be analyzed at once. Set 'Analyse fish no.' to the fish you would like to process.
▲ **CRITICAL STEP** Do not select an integer larger than the number of fish originally recorded.
6 (Optional) If needed, update the path. If no recording is ongoing, enter the path of a data file saved earlier under 'Update data file path' in the 'Eye detection & recording' panel. If you want to inspect the last file saved during this session, do not change the path. If a recording is currently ongoing, you can review the data acquired so far. To do so, do not change the path.
7 Start the analysis. If you wish to analyze all phases at once, click on 'analyse all phases'. Each phase will be analyzed separately, and its results will be saved to a separate file. If you wish to analyze only the single phase selected in step 4 of this box, click on 'analyse phase no.'. The content of the result files is explained in Tables 4 and 5.

Data analysis via GUI ● Timing variable

35 After recording eye traces during oculomotor experiments, perform ad hoc data analysis (Box 3) to determine the OKR gain and saccade counts. The content of the result files created during data analysis is explained in Tables 4 and 5.

Troubleshooting

Troubleshooting advice can be found in Table 6.

Table 6 | Troubleshooting table

Step	Problem	Possible reason	Solution
6	The nilvaiu.dll file is missing	The DAQmx drivers are missing	Install the required NI DAQmx drivers (Equipment)
	The Get Image2.vi file is missing	The IMAQdx components are missing	Update NI Vision Acquisition to v2015 f1, or an even more recent version. Or, if you own a licensed copy of the NI Vision Development Module or its run-time engine, update those instead
	The settings buttons are grayed out	The program is already running in the selected operating mode	Click on 'Stop', and then click the white arrow again to restart. We recommend using 'Run Continuously' (circular arrows) to quickly restart the program when analyzing multiple files. If the orange 'Stop' button is unavailable, click on the red 'Abort' button in the top-left corner of the window
12	An 'IMAQ Image not Found' or 'Exception occurred in IC. ICImaging Control' error is shown	The camera is not connected	As opposed to ZebEyeTrack Light, ZebEyeTrack cannot run without a camera connected. Connect any compatible USB camera and restart ZebEyeTrack
	A camera timeout is indicated	Access to the camera is denied by a virus scanner	Add ZebEyeTrack Light to the list of trusted programs in your virus protection software
	An 'Invalid camera session' error is shown	This can be the result of an intermittent hardware connection	Connect the camera to a different USB port and restart the program. Exchange USB cables and restart the program.
	An 'Undefined function or variable' error is shown	The MATLAB path cannot be initialized	Close all programs. Restart with administrator privileges
	An 'Undefined function "Screen" error is shown	The Psychtoolbox is not installed, or the user lacks permissions	If you do not need Psychtoolbox, just deactivate 'use computer graphics' on the settings panel. Otherwise, install Psychtoolbox for free. If the error persists, run ZebEyeTrack with administrator privileges

Table continued

Troubleshooting table (continued)

Step	Problem	Possible reason	Solution
	An 'Error using "Screen"' error is shown	The stimulus screens are missing	On the settings tab, navigate to the box named 'Psychtoolbox visual stimuli'. Click on 'advanced settings'. Make sure 'first display ID' and 'second display ID' are the same as those assigned by Microsoft Windows (to check, right-click on Windows Desktop, then click on 'Screen resolution'). If no screens are connected, deactivate 'use computer graphics'
	An 'Incompatible image type' error is shown	The file type is not supported	Convert the video into uncompressed 8-bit .avi format using external software. Restart ZebEyeTrack and select the new file. If compression is required, Motion JPEG is preferred
	An 'Error opening AVI file. (...) codec (...) might not have been found.' error is shown	The required codec is missing	ZebEyeTrack supports several compressed and uncompressed .avi video formats: Motion JPEG, FF Video Codec 1, Y800 Uncompressed, and YUV 4:2:0 Planar. Virtually any video file can be converted to any one of these formats. To do so, process your compressed or uncompressed video using free software such as AnyVideoConverter. Make sure never to alter its frame rate during conversion
14	The video fails to load	The file type is not supported	In the case that your own videos cannot be loaded with ZebEyeTrack Light, you can try to open them with the freeware ImageJ/Fiji[38] (Table 2, no. 21), convert them to 8-bit grayscale or RGB image type, and save them as .avi files without compression. These files should now work with ZebEyeTrack Light. We have successfully used non-compressed AVIs saved as AVI v1.0 or as OpenDML AVI v2.02 (no codec: DIB(_RGB), BI_RGB Raw Bitmap). If compression is required, Motion JPEG is preferred
	The displayed video frame does not update	This is due to an unknown LabVIEW error	During live recording, the grayscale image is active only while the pointer of the line tool is clicked in the image and otherwise stops again. Shutting down LabVIEW and restarting it eliminates this problem. This error has not been observed yet in the ZebEyeTrack version presented here
	The video is shown for some time, then an 'Error reading AVI frame' error is shown	This can be the result of a video conversion artifact when altering the frame rate	When converting a video file for use with ZebEyeTrack, its frame rate may accidentally be altered. Convert the original file again, this time maintaining the original frame rate. Reconverting an already-converted file is unlikely to fix the problem
	The live video suddenly freezes, then LabVIEW stops responding	This can be due to an intermittent hardware connection	Connect the camera to a different USB port and restart the program. Exchange USB cables and restart the program
	In live video, fish appear too small	A negative magnification was used	Using the zoom tool and shift-clicking, it is possible to zoom out beyond 1× magnification. Zoom back in by clicking without holding down the shift key
	The sampling rate of eye tracking is too low	On-access virus scanning is activated	Deactivate on-access scanning in your antivirus software
		This can be due to a very high number of particles	Adjust the illumination settings and the sample in the Petri dish in order to minimize the number of particles visible in the binary red/black image
17	The body axes are not saved to the file	The file or path name has changed after setting the body axes	If you change the file path in Step 7 or 13 without subsequently defining new body axes, the previous body axes will be kept in memory, and eye tracking will still work. However, these axes will not be saved to a new .txt file under the new path. If you want the axes to be saved again under the new name, simply click on 'save axis' again after renaming
	No eyes are being detected	The body axis was not saved	After defining the body axis, hold down 'save body axis' for 1 s. Make sure that the number displayed in the 'current body axis' box next to it is updated; this indicates proper saving
19	No video is displayed on the debugging panel	The Image Processing Toolbox for MATLAB is missing	This toolbox, as well as Step 19, is entirely optional. Install the toolbox or skip this step
	The left and right eye positions are swapped	The body axis is inverted	Redefine the body axis. Click first on the midpoint between the eyes and second on the caudal part of the midline, not the other way around
		The fish are filmed from below	The apparent left eye on the screen is in fact the right eye of the fish; with appropriate corrections before or after data analysis, recordings can be used nonetheless. In particular, the sign of eye positions may be swapped (an increase now indicates rightward motion)
		The camera image is inverted	Even when filming from above, certain optical setups may invert the image before it is recorded by the camera. If so, follow the instructions for fish filmed from below

Table continued

Troubleshooting table (continued)

Step	Problem	Possible reason	Solution
	The sign of the eye positions is swapped	The fish is filmed from below	See 'The left and right eye positions are swapped' entry above
	An unexplained sudden change in angular eye position is observed	This can be an effect of struggle motion	After recording, open the *.txt data file. Look for the columns containing the Cartesian coordinates of the center of mass of each eye (Table 1, Supplementary Table 1). If the event was a struggle, both angular and Cartesian eye position should change simultaneously. Improve the embedding method and/or repeat the recording
21	A negative analog output is shown and written to the data file	The saccade marker is ambiguous	To mark saccades in the data file, the value '−4.321' is generated for all output channels. In the case that your equipment supports a negative output, open the block diagram of SaccadeTrigger.vi, unwire the constant '−4.321', and rewire the neighboring constant '0'
	The analog output appears on an unexpected channel	The saccade marker is ambiguous	Negative output is sent to all three possible output channels, not just the one selected by the user. To change, unwire the undesired channels in SaccadeTrigger.vi (see above)
22	Zero entries are shown for fish eye position	This is due to an unknown (particle analysis) error	Sporadically, fish eye-position data cannot be measured (zero entries) while the code runs fine otherwise. This will occur for only one or two eyes, not for all eyes in the recording. The problem is associated with the angle of the eye relative to the camera chip, as rotating the CCD camera slightly always eliminated this problem in our experience
	Unexpected data points, and/or rows of unusual length, at the beginning or end of a recorded .txt file	The save path was not updated	Manually remove the rows containing erroneous data points from the end of the file before further analysis
25A(i), 25B(ii)	The temporal frequencies <0.5 cycles/s are presented incorrectly	The program sets these to 0 cycles/s	Only temporal frequencies ≥0.5 cycles/s can be used. A temporal frequency of 0 is represented as a temporal frequency that will contain exactly one stimulus frame given the phase duration; the program will automatically update a value of 0 cycles/s to the needed value
25A(iii), 25B(ix)	Too few scheduled pauses occur	The schedule starts immediately	The schedule begins upon button click, even if the stimulus has not been started yet. Activate the stimulus first and pauses second
25A(iii), 25B(ix)	The post-saccade pause cannot be set to <0.9 s	This is due to the current software design	This will be fixed in a later release
25C(ii)	The screens flicker when updating the stimulus	This is due to overlapping presentation	Do not click on 'update parameters' while the old stimulus is still running. Wait for the stimulus to end, or deactivate 'present stimulus'. Then, click on 'update parameters'
29	Sampling rate is too low	A video is recorded in parallel	To increase the sampling rate, avoid saving the video to a file while acquiring eye-tracking data. Deactivate 'record video' on the 'Eye detection & recording' tab
35	Error 1129 is shown: 'You cannot assign the same numeric value...'	The data file is corrupted	This is caused by data files with rows of unequal length or by invalid numerical entries in the final column. Manually restore formatting using Table 1 or Supplementary Table 1
	Eye-position data appear erratic	The number of fish selected for analysis exceeded the number in the data file	The post hoc analysis tool does not check whether the number fish you selected for analysis exceeds the total number of fish in the data file. If so, neighboring columns containing other data will be read out instead (Supplementary Table 1)
	An error was displayed when loading the data file, obvious saccades are not detected, or excess saccades are detected	System settings for decimal separator interfere with the algorithm	ZebEyeTrack expects numbers to be formatted with points as decimal separators (e.g., ½ = 0.5), and commas only for optional digit grouping (e.g., 1,000 = 1,000.00). Use editors such as Microsoft Notepad to replace all erroneous symbols (e.g., Ctrl+H), and change system settings ahead of future recordings (i.e., 'Regions' setting on Windows control panel)
	Some saccades are counted twice	These saccades stretch across multiple time steps	Saccade detection in ZebEyeTrack is based on a velocity threshold only. Adjust the analysis parameters. If the problem persists, remove excess saccades from result files, either manually or with custom code
	The slow-phase slope is detected as zero	The saccade is detected near the end of a recording	If a saccade is detected just before the end of a stimulus phase, the time period of the subsequent slow-phase motion may fall outside the time range covered by the stimulus phase and thus incorrectly be measured as having a slope of zero. Manually discard the result, or shorten the data file before analyzing

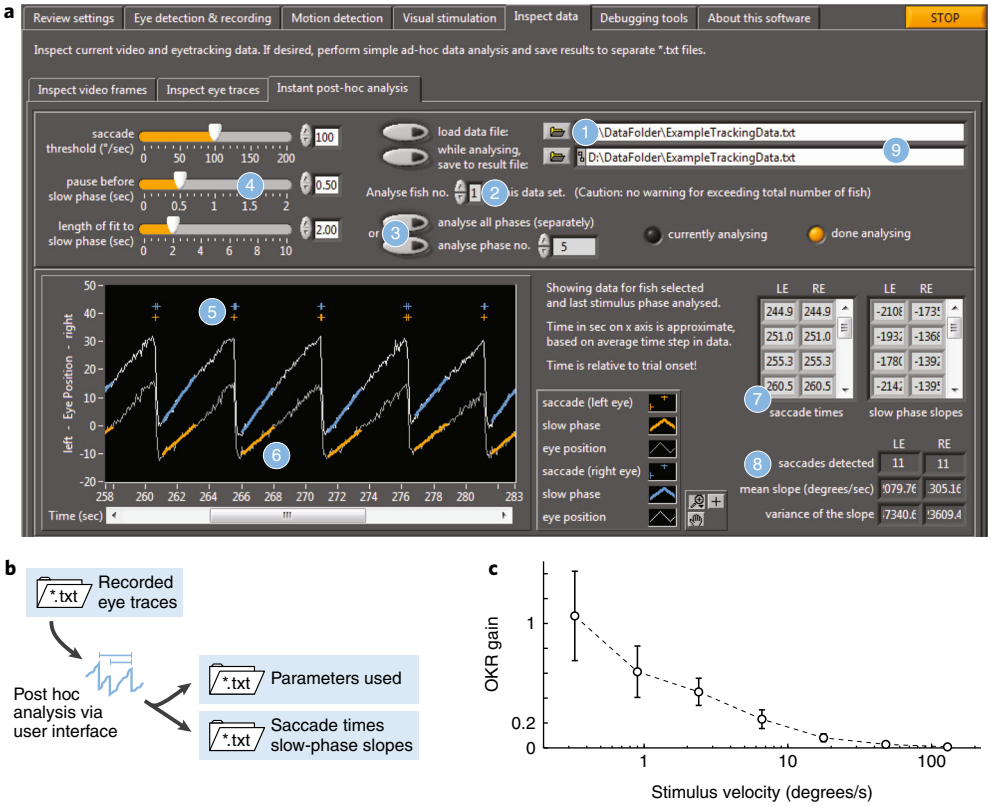

Fig. 7 | Data analysis in ZebEyeTrack. a, As an alternative or complement to existing data-analysis pipelines, previously acquired tracking data (1) can be analyzed through the graphical user interface. For up to six fish recorded in parallel (2), eye traces from both eyes and all stimuli can be processed with a single click, one fish at a time (3), but results are still computed separately for each eye and each stimulus phase. On the basis of user-defined parameters (4), saccades are detected (5), and the slope of ensuing slow-phase eye movements is estimated (6). Along with saccade times (7) and the total number of saccades (8), these results are automatically saved to separate files (9). For further details, see 'Post hoc analysis' in the Procedure. **b**, Using this interface, previously recorded eye-tracking data files are read in, and the results of data analysis are saved into a separate .txt output file for each stimulus phase presented (see Table 5 for the content of these result files). Stimulus phases are automatically detected from the original data file, and user-selected analysis parameters as well as meta information, such as the number of saccades per phase, are saved to a single .txt file for later reference (see Table 4). **c**, The experimenter can further process .txt result files to generate response curves. Here, five fish were recorded simultaneously at 7 d.p.f. to determine the OKR gain of their slow-phase eye movements as a function of stimulus velocity. The slopes of their slow phases were pooled, such that each median value shown reflects between 22 and 128 individual data points. OKR gain is equal to the slope of slow phases, as determined by ZebEyeTrack, subsequently divided by stimulus velocity, and is thus unitless. Stimulus spatial frequency was 30 degrees/cycle or 0.033 cycles/degree. Error bars indicate standard error of the mean. Appropriate regulatory board permission was obtained before zebrafish experiments.

Timing

Steps 1–4, installation and setup: variable
Step 5, immobilization of zebrafish larvae: 5–10 min
Steps 6–13, software configuration: 1 min
Steps 14–19, eye detection: 2 min
Steps 20 and 21, (optional) analog output: 1 min
Steps 22–24, eye tracking and recording: variable (typically 1–15 min)
Steps 25–28, (optional) stimulus design and presentation: 5–10 min
Steps 29 and 30, (optional) recording of a video: variable
Steps 31–33, inspection of the latest data file or video: variable
Step 34, ending the experiment: instantaneous
Step 35, data analysis via GUI: variable (0–2 min per stimulus phase)
Box 1, eye tracking for an existing video: 5 min + length of the video

Box 2, general motion detection: 2 min
Box 3, data analysis via the GUI: 0–2 min per stimulus phase

Anticipated results

With ZebEyeTrack, experimenters can track the horizontal eye positions of up to six larval zebrafish at a resolution of 0.1° (Figs. 2 and 6) and a sampling rate of up to 60 f.p.s. The slow phase and the quick phase (saccade) of the OKR can be analyzed to retrieve saccade times, counts, and slow-phase eye velocity (Fig. 7). The results of ZebEyeTrack post hoc analysis can then be used to generate fish-averaged stimulus response curves for a particular visual parameter such as stimulus velocity (Fig. 7c). As detailed in the Introduction, ZebEyeTrack can be helpful for many other applications, such as optogenetic experiments or calcium imaging (Supplementary Fig. 2, parts of which were adapted from previous work[37]). The online supplement includes two example videos (Supplementary Videos 1 and 2), along with the eye-tracking data files that can be recorded from them (Supplementary Data 1 and 2). To limit file size, these videos are compressed using Motion JPEG compression. The uncompressed videos are available on our website (http://www.zebeyetrack.org/videos/video_1.avi and http://www.zebeyetrack.org/videos/video_2.avi). The supplementary information also features several example stimulus protocols for both stimuli presented on custom LED arrays (Supplementary Data 3 and 4) and those presented on regular computer screens or via video projection (Supplementary Data 5). To try out post hoc analysis via the user interface, a sample file with raw eye-tracking data (Supplementary Data 6), as well as some of the expected result files after analysis (Supplementary Data 7–9) are included.

References

1. van Alphen, A. M., Stahl, J. S. & De Zeeuw, C. I. The dynamic characteristics of the mouse horizontal vestibulo-ocular and optokinetic response. *Brain Res.* **890**, 296–305 (2001).
2. Faulstich, B. M., Onori, K. A. & du Lac, S. Comparison of plasticity and development of mouse optokinetic and vestibulo-ocular reflexes suggests differential gain control mechanisms. *Vision Res.* **44**, 3419–3427 (2004).
3. Cahill, H. & Nathans, J. The optokinetic reflex as a tool for quantitative analyses of nervous system function in mice: application to genetic and drug-induced variation. *PLoS ONE* **3**, e2055 (2008).
4. Land, M. F. Eye movements of vertebrates and their relation to eye form and function. *J. Comp. Physiol. A* **201**, 195–214 (2015).
5. Easter, S. S. Jr & Nicola, G. N. The development of vision in the zebrafish (*Danio rerio*). *Dev. Biol.* **180**, 646–663 (1996).
6. Roeser, T. & Baier, H. Visuomotor behaviors in larval zebrafish after GFP-guided laser ablation of the optic tectum. *J. Neurosci.* **23**, 3726–3734 (2003).
7. Brockerhoff, S. E. et al. A behavioral screen for isolating zebrafish mutants with visual system defects. *Proc. Natl. Acad. Sci. USA* **92**, 10545–10549 (1995).
8. Neuhauss, S. C. F. et al. Genetic disorders of vision revealed by a behavioral screen of 400 essential loci in zebrafish. *J. Neurosci.* **19**, 8603–8615 (1999).
9. Cameron, D. J. et al. The optokinetic response as a quantitative measure of visual acuity on zebrafish. *J. Vis. Exp.* **80**, e50832 (2013).
10. Distler, C., Vital-Durand, F., Korte, R., Korbmacher, H. & Hoffmann, K.-P. Development of the optokinetic system in macaque monkeys. *Vision Res.* **39**, 3909–3919 (1999).
11. Distler, C. & Hoffmann, K.-P. Development of the optokinetic response in macaques: a comparison to cat and man. *Ann. N. Y Acad. Sci.* **1004**, 10–18 (2003).
12. Muto, A. et al. Forward genetic analysis of visual behavior in zebrafish. *PLoS Genet.* **1**, 0575–0588 (2005).
13. Brockerhoff, S. E. Measuring the optokinetic response of zebrafish larvae. *Nat. Protoc.* **1**, 2448–2451 (2006).
14. Schoonheim, P. J., Arrenberg, A. B., Del Bene, F. & Baier, H. Optogenetic localization and genetic perturbation of saccade-generating neurons in zebrafish. *J. Neurosci.* **30**, 7111–7120 (2010).
15. Bianco, I. H. et al. The tangential nucleus controls a gravito-inertial vestibulo-ocular reflex. *Curr. Biol.* **22**, 1285–1295 (2012).
16. Portugues, R., Feierstein, C. E., Engert, F. & Orger, M. B. Whole-brain activity maps reveal stereotyped, distributed networks for visuomotor behavior. *Neuron* **81**, 1328–1343 (2014).
17. Kubo, F. et al. Functional architecture of an optic flow-responsive area that drives horizontal eye movements in zebrafish. *Neuron* **81**, 1344–1359 (2014).
18. Mueller, K. P. & Neuhauss, S. C. F. Quantitative measurements of the optokinetic response in adult fish. *J. Neurosci. Methods* **186**, 29–34 (2010).
19. Mueller, K. P., Schnaedelbach, O. D. R., Russig, H. D. & Neuhauss, S. C. F. VisioTracker, an innovative automated approach to oculomotor analysis. *J. Vis. Exp.* **56**, e3556 (2011).

20. Arrenberg, A. B. in *Zebrafish: Methods and Protocols (Methods in Molecular Biology 1451), Fiber oOptic-based Photostimulation of Larval Zebrafish* 343–354 (Springer Science+Business Media 2016).
21. Arrenberg, A., Del Bene, F. & Baier, H. Optical control of zebrafish behavior with halorhodopsin. *Proc. Natl. Acad. Sci. USA* **106**, 17968–73 (2009).
22. Gonçalves, P. J., Arrenberg, A. B., Hablitzel, B., Baier, H. & Machens, C. K. Optogenetic perturbations reveal the dynamics of an oculomotor integrator. *Front. Neural Circuits* **8**, 1–22 (2014).
23. Miri, A. et al. Spatial gradients and multidimensional dynamics in a neural integrator circuit. *Nat. Neurosci.* **14**, 1150–1159 (2011).
24. Reinig, S., Driever, W. & Arrenberg, A. B. The descending diencephalic dopamine system is tuned to sensory stimuli. *Curr. Biol.* **27**, 1–16 (2017).
25. Thiele, T. R., Donovan, J. C. & Baier, H. Descending control of swim posture by a midbrain nucleus in zebrafish. *Neuron* **83**, 1–13 (2014).
26. Clark, D. *Visual Responses in the Developing Zebrafish (Brachydanio rerio)*. PhD thesis (University of Oregon Press, 1981).
27. Huber-Reggi, S. P., Mueller, K. P. & Neuhauss, S. C. in *Retinal Degeneration: Methods and Protocols (Methods in Molecular Biology 935)* (eds. Weber, B. H. & Langmann, T.) 139–160 (Springer Science+Business Media, 2013).
28. Kretschmer, F., Kretschmer, V., Kunze, V. P. & Kretzberg, J. OMR-Arena: automated measurement and stimulation system to determine mouse visual thresholds based on optomotor responses. *PLoS ONE* **8**, e78058 (2013).
29. Miri, A., Daie, K., Burdine, R. D., Aksay, E. & Tank, D. W. Regression-based identification of behavior-encoding neurons during large-scale optical imaging of neural activity at cellular resolution. *J. Neurophysiol.* **105**, 964–980 (2011).
30. Brysch, C., Leyden, C. & Arrenberg, A. An investigation of the neuronal tuning to horizontal eye movements in the oculomotor system of larval zebrafish. Program no. 150. 14. *Neuroscience Meeting Planner: Society for Neuroscience* (Washington, DC, 2017).
31. Reiff, D. F., Plett, J., Mank, M., Griesbeck, O. & Borst, A. Visualizing retinotopic half-wave rectified input to the motion detection circuitry of *Drosophila*. *Nat. Neurosci.* **13**, 973–978 (2010).
32. Branchek, T. The development of photoreceptors in the zebrafish, Brachydanio rerio. II. function. *J. Comp. Neurol.* **224**, 116–122 (1984).
33. Brainard, D. The psychophysics toolbox. *Spat. Vis.* **10**, 433–436 (1997).
34. Pelli, D. The VideoToolbox software for visual psychophysics: transforming numbers into movies. *Spatial Vision* **10**, 437–442 (1997).
35. Kleiner, M., Brainard, D. & Pelli, D. What's new in Psychtoolbox-3? *Perception 36 ECVP Abstract Supplement* (Arezzo, Italy, 2007).
36. To, L., Woods, R. L., Goldstein, R. B. & Peli, E. Psychophysical contrast calibration. *Vis. Res.* **90**, 15–24 (2013).
37. Hablitzel, B. Diploma Thesis (Freiburg University, 2012).
38. Schindelin, J. et al. Fiji: an open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012).
39. Lopes, G. et al. Bonsai: an event-based framework for processing and controlling data streams. *Front. Neuroinform* **9**, 7 (2015).

Acknowledgements

ZebEyeTrack is based on a precursory application that was developed by A.B.A. in the laboratory of H. Baier (UCSF) and refined in the laboratory of W. Driever (Freiburg University). Parts of the MATLAB Psychtoolbox scripts for running visual stimuli were based on pre-existing scripts provided by M.B. Orger (Baier Lab, UCSF). We thank L. Ziv-Strasser, D. Strasser, and J. Huisken for teaching A.B.A. how to use LabVIEW at UCSF. We thank B. Hablitzel (Freiburg University) for helping to develop the custom LED arena, a pilot benchmarking of eye-tracking performance, and for technical assistance. We also thank G. Lopes (Kampff Lab, UCL) for helpful conversations on software development. At the Arrenberg lab, C. Brysch tested and provided feedback on our software; S. Buss and R. Meier assisted with preparations for software testing and benchmarking. This work was supported by Deutsche Forschungsgemeinschaft (DFG) grants EXC307 (CIN-Interdisciplinary Centre for Integrative Neuroscience) and INST 37/967-1 FUGG, as well as a Juniorprofessor programme grant from the Ministry of Science, Research, and the Arts of the State of Baden-Württemberg (MWK).

Author contributions

A.B.A. conceived the software and experiments and wrote a precursory version of the software. F.A.D., C.L., and A.B.A. wrote the code for ZebEyeTrack. F.A.D. and A.v.D. ensured compatibility across operating systems. A.v.D. set up software repositories. F.A.D and A.B.A. wrote the manuscript. F.A.D. designed the user interface, performed experiments, and created the figures.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41596-018-0002-0.

Reprints and permission information is available at http://www.nature.com/reprints

Correspondence and requests for materials should be addressed to A.B.A.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Published online: 9 July 2018

Related links

Functional architecture of an optic flow-responsive area that drives horizontal eye movements in zebrafish: https://doi.org/10.1016/j.neuron.2014.02.043
Optogenetic Localization and Genetic Perturbation of Saccade-Generating Neurons in Zebrafish: https://doi.org/10.1523/JNEUROSCI.5193-09.2010