BonVision – an open-source software to create and control visual environments

Gonçalo Lopes[1], Karolina Farrell[2,‡], Edward A. B. Horrocks[2,‡], Chi-Yu Lee[2,‡], Mai M. Morimoto[2,‡], Tomaso Muzzu[2,‡], Amalia Papanikolaou[2,‡], Fabio R. Rodrigues[2,‡], Thomas Wheatcroft[2,‡], Stefano Zucca[2,‡], Samuel G. Solomon[2,*], Aman B. Saleem[2,*]

[1]NeuroGEARS Ltd., London; [2]UCL Institute of Behavioural Neuroscience, Department of Experimental Psychology, University College London, London, WC1H 0AP.
[‡]These authors are listed alphabetically.
*These authors jointly supervised this work.

Real-time rendering of closed-loop visual environments is necessary for next-generation understanding of brain function and behaviour, but is prohibitively difficult for non-experts to implement and is limited to few laboratories worldwide. We developed BonVision as an easy-to-use software for display of virtual or augmented reality as well as standard visual stimuli. As the architecture is based on the open-source Bonsai graphical programming language, BonVision benefits from Bonsai's native integration with experimental hardware. BonVision therefore enables easy implementation of closed-loop experiments and communication with behavioural and physiological measurement and manipulation devices, while being open-source, easy to install, and able to run on standard personal computers.

Understanding behaviour and its underlying neural mechanisms requires the ability to construct and control environments that immerse animals, including humans, in complex naturalistic environments that are responsive to their actions. Gaming-driven advances in computation and rendering have driven the development of immersive closed-loop visual environments, but these new platforms are not readily amenable to traditional research paradigms. For example, they do not specify an image in egocentric units (of visual angle), sacrifice precise control of a visual display, lack transparent interaction with external hardware and require advanced computational training to implement.

Most vision research has been performed in non-immersive environments with standard two-dimensional visual stimuli, such as grating or dot stimuli, using established platforms including PsychToolbox [1] or PsychoPy [2,3]. Pioneering efforts to bring gaming-driven advances to neuroscience research have provided new platforms for closed-loop visual stimulus generation: STYTRA provides visual stimuli for larval zebrafish in python [4], ratCAVE is a specialised augmented reality system for rodents in python [5], FreemoVR provides virtual reality in Ubuntu/Linux [6], and ViRMEn provides virtual reality in Matlab [7]. However, these new platforms lack the generalised frameworks needed to specify or present standard visual stimuli.

We developed BonVision, a software package for the Bonsai visual programming language [8] that can generate and display well-defined visual stimuli in closed-loop 2D and 3D environments. BonVision extends Bonsai by providing pre-built GPU shaders and resources for most stimuli used in vision research, along with an accessible, modular interface for composing stimuli and designing experiments.

One important challenge in the design of visual experiments is the need for precise specification of visual scenes, either in degrees of visual field

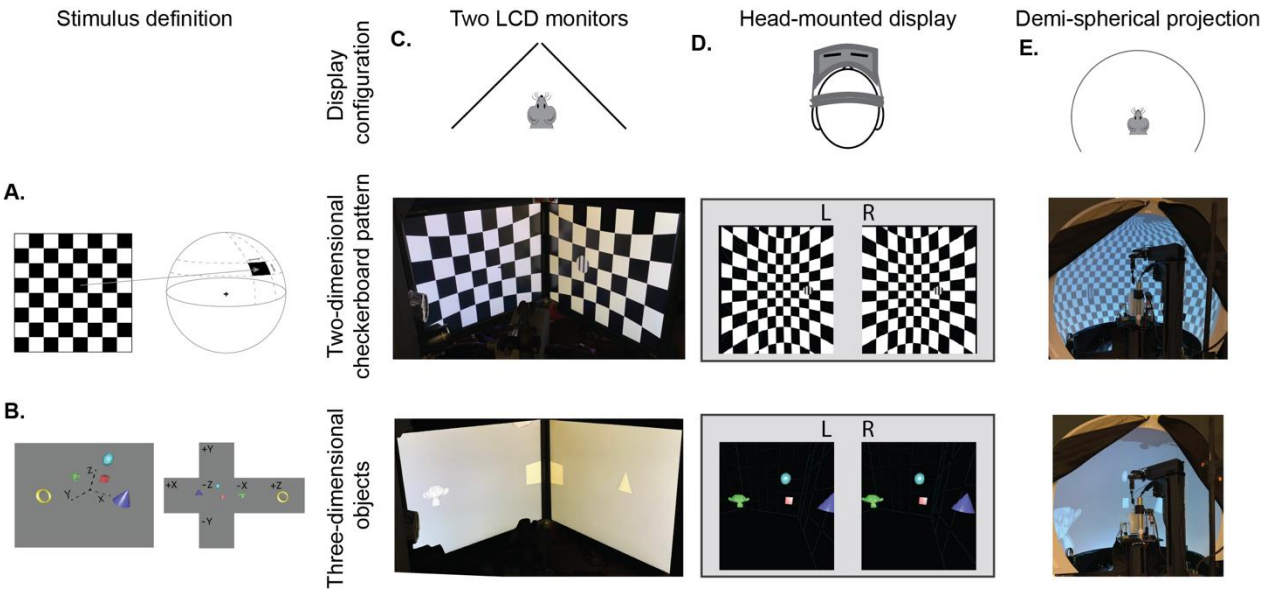


Figure 1: BonVision's adaptable display and render configurations. A. Illustration of how two-dimensional textures are generated in BonVision using Mercator projection for sphere mapping, with elevation as latitude and azimuth as longitude. **B.** Three-dimensional objects were placed at the appropriate positions and the visual environment was rendered using cube-mapping. **C-E.** Examples of the same two stimuli, a checkerboard + grating (middle row) or four three-dimensional objects (bottom row), displayed in different experimental configurations (top row): two angled LCD monitors (**C**), a head-mounted display (**D**), and demi-spherical dome (**E**).

(for 2D scenes), or in metric units (for 3D scenes). For instance, even for a single computer monitor, the visual angle that is subtended by a pixel on the monitor depends on the viewing distance between the subject and the pixel, which varies with the position of that pixel on the monitor. While compensation for this spatial distortion is achievable, it is rarely done. This is particularly problematic when the same experiment needs to be ported between experimental apparatus. For example, this might be required when replicating an experiment that previously used a standard computer monitor for image display to a new set-up that uses a curved display surface or multiple monitors angled around the subject. Each of these conversions requires recalculating the view of the scene that is provided by the display, and therefore how the scene is rendered on that display. BonVision was designed with the need for this flexibility in mind, by making configuration and calibration of the visual display(s) independent of scene specification. Consequently, the same experimental code could be run on any calibrated apparatus, and once an apparatus is calibrated it can be used to reproduce any BonVision experiment. Enforcing the separation between scene specification and display environment has the additional benefit that BonVision unites 2D and 3D environments into the same framework, which can be used for display of standard visual stimuli and virtual or augmented reality.

Method & Results

To allow specification of standard visual stimuli in terms of degrees of visual angle, we first mapped the retino-centric coordinate frame (i.e. degrees of the visual field) to the surface of a 3D sphere using Mercator projection (Fig 1A, Suppl. Fig 1). The resulting sphere could therefore be rendered onto displays in the same way as any other 3D environment.

We used "cube mapping" to specify the projection of 3D environments onto an experimental subject. Here, the 6 faces of a cube, each covering a 90º field of view volume, are combined to describe an entire 360º visual

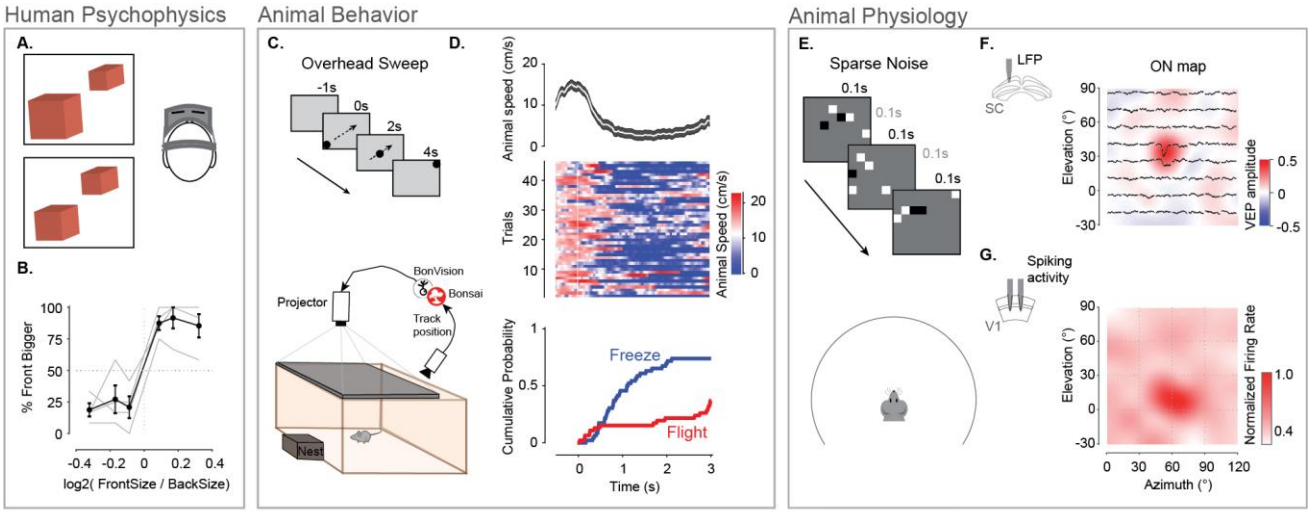


Figure 2: Validation of BonVision in different vision research experiments. **A.** Two cubes were presented at different distances in a virtual environment through a head-mounted display to human subjects. Subjects had to report which side had the larger cube: left or right. **B.** Subjects predominantly reported the larger object to be bigger, with a slight bias to report that the object in front was bigger. **C.** BonVision used to present visual stimuli overhead while an animal was free to explore an environment (which included a refuge). Movements of the animal were tracked using standard tracking nodes in Bonsai. The stimulus was a small dot (5^0 diameter) moving across the projected surface over several seconds. **D.** The speed of the animal across different trials, aligned to the time of stimulus appearance. **E.** BonVision used to present 'sparse noise' visual stimuli via projection onto a demi-spherical dome. **F-G.** Receptive field maps obtained using this sparse noise stimulus in recordings of local field potential in the superior colliculus (**F**), or spiking activity in the primary visual cortex (**G**) of mouse.

environment (Fig 1B). A display device then becomes a window into that environment, and each pixel on the display specifies a vector from the subject through that window. The vector links pixels in the display to pixels in the cubemap, thereby rendering the corresponding portion of the visual field onto the display.

Our approach has the advantage that the visual environment is defined independently of display hardware, which allows us to consider visual displays as arbitrary windows between the subject and the 3D environment (Fig 1C-E, Suppl Fig 1, 2). These windows can be created independently for each experimental apparatus without changing the preceding specification of the visual scene, or the experimental design (Fig 1C-E, Suppl Fig 1, 2). Consequently, BonVision generates visual environments and experiments that can be trivially replicated on very different display devices, including multiple monitors, curved projection surfaces, and head-mounted displays (Fig 1C-E).

To establish the effectiveness of BonVision, we tested it in several experimental paradigms: human psychophysics, animal behaviour and neurophysiology. To test the perceptual quality of the system we asked human subjects to discriminate the size of objects presented at different depths in a head-mounted display (HTC Vive)[9]. On each trial, subjects identified the larger of two non-overlapping cubes that were placed at different virtual depths (Fig 2A-B). The display was updated in closed-loop to allow subjects to alter their viewpoint by moving their head. Distinguishing objects of the same retinal size required subjects to use depth-dependent cues [10], and we found that all subjects were able to identify which cube was larger (Fig 2B). These results suggest that BonVision provides

sufficient rendering performance to support human depth perception.

As an additional test, we asked if BonVision could trigger instinctive avoidance behaviours in mice (Fig 2C-D). We displayed a small black dot slowly sweeping across the overhead visual field, which is known to primarily elicit a freezing response in mice [11] (Fig 2C). Visual stimuli presented in BonVision elicited similar behavioural responses to those previously described (Fig 2D). This experiment also illustrates an advantage of leveraging Bonsai: we included pre-existing Bonsai video analysis nodes in the same code fragment to automatically track the position of the animal. Bonsai has a large and growing number of established protocols for online analysis of experimental data from sensors including video cameras, and BonVision is immediately compatible with them. BonVision can therefore can easily integrate information from different sources in real-time, within the same workflow.

To establish whether BonVision provides a sufficient framework for standard visual neurophysiology experiments, where there is higher demand for timing accuracy, we mapped the receptive fields of neurons early in the visual pathway [12], in the primary visual cortex and superior colliculus. The stimulus was 5 small black or white squares briefly (0.1s) presented at random locations (Fig 2E). We chose this stimulus because the ability to extract receptive fields from neural responses is sensitive to the timing accuracy of the visual stimulus and the synchronisation of the stimulus with the recorded data, meaning that errors in timing would prevent the identification of receptive fields. Using BonVision, we recovered clear receptive fields from electrophysiological measurements [13] in the superior colliculus and primary visual cortex of awake mouse (Fig 2F-G).

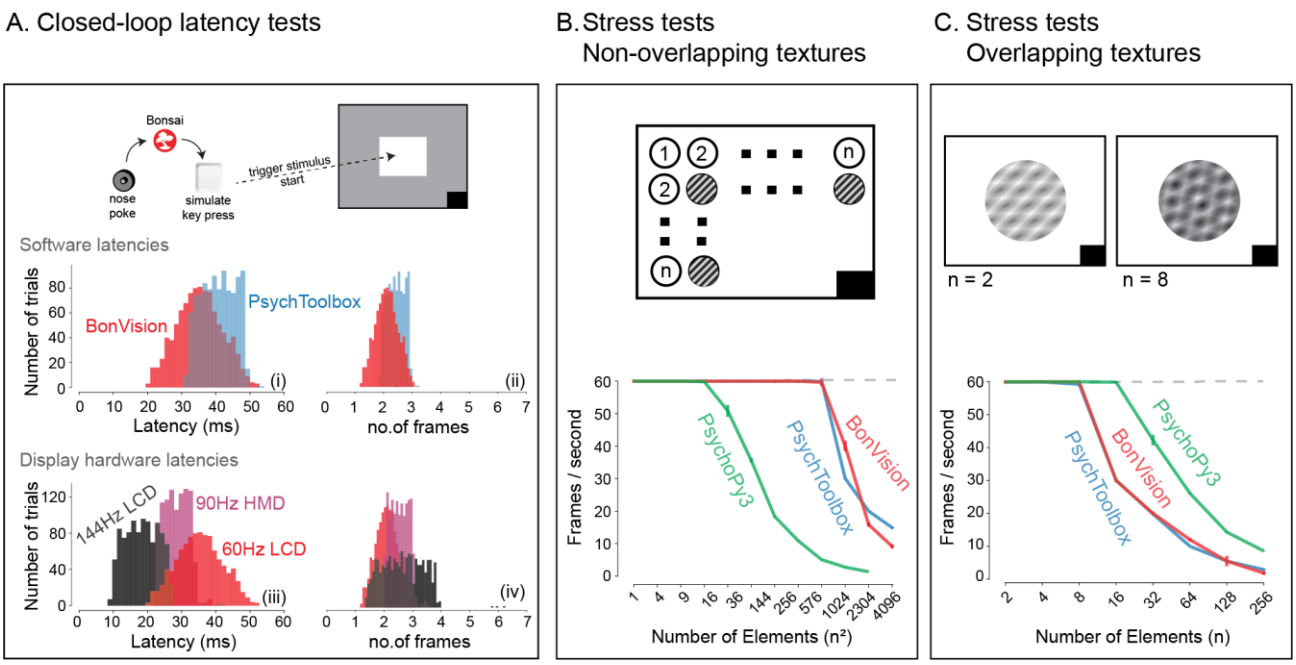


Figure 3: Closed-loop latency and benchmarks. A. Latency between sending a command (virtual key press) and updating the display (measured using a photodiode). (A.i - A.ii) Latency depended on the frame rate of the display, updating stimuli with a delay of 1-3 frames. (A.iii - A.iv). **B-C.** Benchmarked performance of BonVision with respect to Psychtoolbox and PsychoPy. **B.** When using non-overlapping textures BonVision and Psychtoolbox could present 576 independent textures without dropping frames, while PsychoPy could present 16. **C.** When using overlapping textures PsychoPy could present 16 textures, while BonVision and Psychtoolbox could present 8 textures without dropping frames.

To establish whether BonVision provides an effective framework for closed-loop stimuli, we measured the delay (latency) between an external event and the presentation of a visual stimulus. To do this, we used a virtual key press as the trigger, which was generated by a separate instantiation of Bonsai and was elicited by the breaking of an LED beam. The delay between beam-break and the virtual key press was less than 2ms, as it should be half the measured round-trip time of 1.58 ± 0.59ms (here and in subsequent reported measurements, ± indicates two standard deviations of the mean, from 1000 repetitions). The virtual key press triggered the presentation of a white square, the appearance of which was monitored by a photodiode, and its output was acquired by the same device that generated the beam-break (HARP; https://www.cf-hw.org/harp). We first measured the closed-loop latency for BonVision when the monitor was refreshed at a rate of 60Hz. We found delays averaged 2.11 ± 0.78 frames (35.26 ± 13.07ms). This latency was slightly shorter than that achieved by PsychToolbox on the same laptop (2.44 ± 0.59 frames, 40.73 ± 9.8ms; Welch's t-test, $p < 10^{-80}$, N=1000).

The overall latency of BonVision was predominantly dictated by the refresh rate of the display device, such that higher frame rate displays yielded lower latency (60Hz: 35.26 ± 13.07ms; 90Hz: 28.45 ± 7.22ms; 144Hz: 18.49 ± 10.1ms; Fig. 3B); that is, the number of frames between the external event and stimulus presentation was fairly constant across frame rate (60Hz: 2.11 ± 0.78 frames; 90Hz: 2.56 ± 0.65 frames; 144Hz: 2.66 ± 1.45 frames; Fig. 3C).

To establish the rendering performance of BonVision relative to other frameworks, we parametrically manipulated the complexity of a stimulus frame, using two methods to benchmark performance (Fig 3C-F). We first measured the number of non-overlapping texture elements that could be rendered within a single frame, without missing ("dropping") frames (we note that we have not tried to optimise code fragments for each framework). We found that BonVision and Psychtoolbox were both able to render up to 576 independent elements at 60Hz, whereas PsychoPy was capable of rendering 36 independent elements. We then measured performance for overlapping textures and found that BonVision and PsychToolbox were both able to render up to 8 overlapping textures, and PsychoPy managed 16. We conclude that the rendering performance of BonVision matches the already high standards achieved by other standard platforms.

BonVision has a growing library of workflows and operators to standardize and ease the construction of new stimuli and virtual environments. For example, it currently has established protocols for defining display positions (Suppl Fig 2), linearizing displays, and mesh-mapping of curved displays (Fig 1E). Bonsai [8] already has a large user base and active developer community, and is now a standard tool for open-source neuroscience research. This established community maximises the long-term sustainability and development of the BonVision platform.

Discussion

We developed a new software, BonVision, that can generate complex 3D environments and 2D, retinotopically defined visual stimuli in the same framework, and separates the specification of these environments from display characteristics. As it is based on the Bonsai programming language, BonVision naturally integrates with a wide range of experimental hardware, which supports real-time modulation of the visual stimulus. The rendering performance of BonVision matches that of other standard visual psychophysics packages, with the added functionality that experiments which closely integrate with hardware can be rapidly prototyped and easily replicated across different display configurations. Being free, open-source and portable, BonVision makes state-of-the-art

tools in vision research accessible to a wider community.

Code availability

BonVision is an open-source software package available to use under the MIT license. It can be downloaded through the Bonsai (bonsai-rx.org) package manager, and the source code is available at: github.com/bonvision/BonVision. Installation instructions, demos and learning tools are available at: bonvision.github.io/.

Acknowledgements

This work was supported by a Wellcome Enrichment award: Open Research (200501/Z/16/A), Sir Henry Dale Fellowship from the Wellcome Trust and Royal Society (200501), Human Science Frontiers Program grant (RGY0076/2018) to A.B.S., an International Collaboration Award (with Adam Kohn) from the Stavros Niarchos Foundation / Research to Prevent Blindness to S.G.S, Medical Research Council grant (R023808), Biotechnology and Biological Sciences Research Council grant (R004765) to S.G.S. and A.B.S.

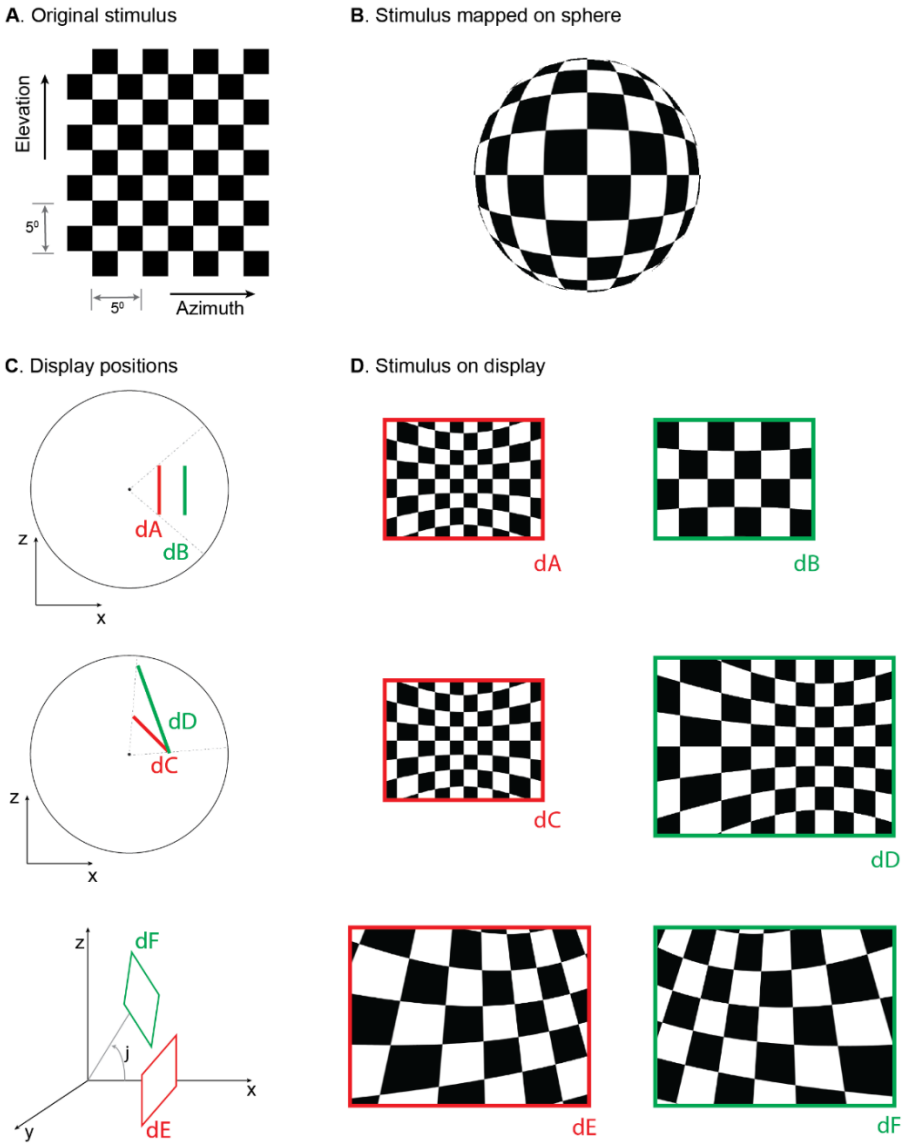
Author Contributions

This work was conceptualized by G.L., S.G.S. and A.B.S., the software was developed by G.L., methodology and validation were by all authors, writing – original draft was by G.L., S.G.S. and A.B.S., and writing – review & editing was by G.L., K.F., M. M. M., T.M., T.W., S.Z., S.G.S. & A.B.S, and supervision and funding acquisition was by S.G.S. and A.B.S.

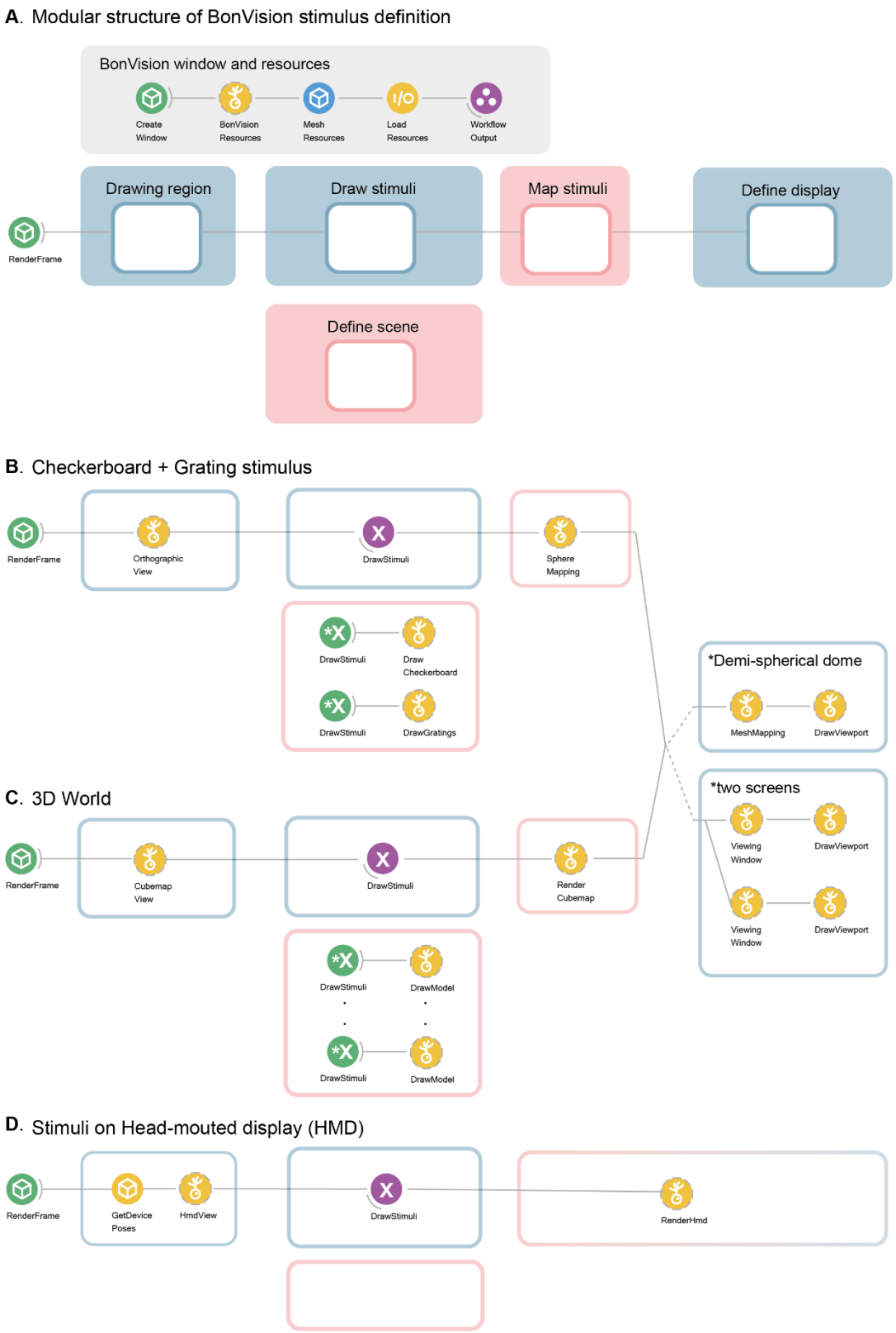
References

1. Brainard DH: **The psychophysics toolbox**. *Spat Vis* 1997, **10**:433.
2. Peirce JW: **PsychoPy-Psychophysics software in Python**. *J Neurosci Methods* 2007, **162**:8–13.
3. Peirce JW: **Generating stimuli for neuroscience using PsychoPy**. *Front Neuroinform* 2009, **2**.
4. Štih V, Petrucco L, Kist AM, Portugues R: **Stytra: An open-source, integrated system for stimulation, tracking and closed-loop behavioral experiments**. *PLOS Comput Biol* 2019, **15**:e1006699.
5. Del Grosso NA, Sirota A: **Ratcave: A 3D graphics python package for cognitive psychology experiments**. *Behav Res Methods* 2019, **51**:2085–2093.
6. Stowers JR, Hofbauer M, Bastien R, Griessner J, Higgins P, Farooqui S, Fischer RM, Nowikovsky K, Haubensak W, Couzin ID, et al.: **Virtual reality for freely moving animals**. *Nat Methods* 2017, **14**:995–1002.
7. Aronov D, Tank DW: **Engagement of Neural Circuits Underlying 2D Spatial Navigation in a Rodent Virtual Reality System**. *Neuron* 2014, **84**:442–456.
8. Lopes G, Bonacchi N, Frazão J, Neto JP, Atallah B V, Soares S, Moreira L, Matias S, Itskov PM, Correia PA, et al.: **Bonsai: an event-based framework for processing and controlling data streams.** *Front Neuroinform* 2015, **9**:7.
9. Scarfe P, Glennerster A: **Using high-fidelity virtual reality to study perception in freely moving observers**. *J Vis* 2015, **15**:3–3.
10. Rolland JP, Gibson W, Ariely D: **Towards Quantifying Depth and Size Perception in Virtual Environments**. *Presence Teleoperators Virtual Environ* 1995, **4**:24–49.
11. De Franceschi G, Vivattanasarn T, Saleem AB, Solomon SG: **Vision Guides Selection of Freeze or Flight Defense Strategies in Mice**. *Curr Biol* 2016, **26**:2150–2154.
12. Yeh CI, Xing D, Williams PE, Shapley RM: **Stimulus ensemble and cortical layer determine V1 spatial receptive fields**. *Proc Natl Acad Sci U S A* 2009, **106**:14652–14657.
13. Siegle JH, López AC, Patel YA, Abramov K, Ohayon S, Voigts J: **Open Ephys: An open-source, plugin-based platform for multichannel electrophysiology**. *J Neural Eng* 2017, **14**:045003.

Supplementary Figures



Supplementary Fig 1: Mapping stimuli onto displays in various positions.
A. Checkerboard stimulus being rendered. **B.** Projection of the stimulus onto a sphere using Mercator projection. **C.** Example display positions (dA-dF) and (**D**) corresponding rendered images.



Supplementary Fig 2: Modular structure of workflow and example workflows.
A. Description of the modules in BonVision workflows that generate stimuli. Every BonVision stimuli includes a module that initiates the render the workflow with "BonVision window and resources". This defines the window parameters in *Create Window* (such as background colour, screen index, VSync), and loads predefined (*BonVision Resources*) and user defined textures (*Texture Resources*, not shown), and 3D meshes (*Mesh Resources*). This is followed by the modules: "Drawing region",

where the visual space covered by the stimuli is defined, which can be the complete visual space, 360º x 360º. “Draw stimuli” and “Define scene” are where the stimulus is defined, “Map Stimuli”, which maps the stimuli into the 3D environment, and “Define display”, where the display devices are defined. **B-C.** Modules that define the checkerboard + grating stimulus (**B.**) shown in the middle row of Fig 1, and 3D world (**C.**) with 5 objects shown in the bottom row of Fig 1. The display device is defined separately and either display can be appended at the end of the workflow. This separation of the display device allows for replication between experimental configurations. **D.** The variants of the modules used to display stimuli on a head-mounted display. The empty region under “Define scene” would be filled by the corresponding nodes in **B** and **C**.

Experimental Methods

Benchmarking

We performed benchmarking to measure latencies and skipped ("dropped") frames. For benchmarks at 60Hz refresh rate we used a standard laptop with the following configuration: Dell Latitude 7480, Intel Core i7-6600U Processor Base with Integrated HD Graphics 520 (Dual Core, 2.6GHz), 16GB RAM. For higher refresh rates we used a gaming laptop ASUS ROG Zephyrus GX501GI, with an Intel Core i7-8750H (6 cores, 2.20GHz), 16GB RAM, equipped with a NVIDIA GeForce GTX 1080. The gaming laptop built-in display refreshes at 144Hz, and for measuring latencies at 90Hz we connected it to a Vive Pro SteamVR head-mounted display (90Hz refresh rate). All tests were run on Windows 10 Pro 64-bit.

To accurately measure the time from input detection to display update, as well as dropped frames detection, we used open-source HARP devices from Champalimaud Research Scientific Hardware Platform. Specifically we used the HARP Behavior device (https://www.cf-hw.org/harp/behavior) to synchronise all measurements with the extensions: Photodiode v2.1 to measure the change of the stimulus on the screen, and Mice poke simple v1.2 as the nose poke device to externally trigger changes. To filter out the infrared noise generated from an internal LED sensor inside the Vive Pro HMD, we positioned an infrared cut-off filter between the internal headset optics and the photodiode.

All benchmark programs and data are available at https://github.com/bonvision/benchmarks.

Animal Experiments

All experiments were performed in accordance with the Animals (Scientific Procedures) Act 1986 (United Kingdom) and Home Office (United Kingdom) approved project and personal licenses. The experiments were approved by the University College London Animal Welfare Ethical Review Board under Project License 70/8637. The mice (C57BL6 wild-type) were group-housed with a maximum of five to a cage, under a 12-hour light/dark cycle. All behavioural and electrophysiological recordings were carried out during the dark phase of the cycle.

Innate Defensive Behaviour

To test innate defensive responses, mice (5 males, C57BL6, 8 weeks old) were placed in a behavioural arena 40cm wide x 40cm long and 33cm high. We positioned a dark rectangular shelter (20cm wide x 10cm long and 10 cm high) outside the arena. Mice could access the shelter through a 10cm wide x 5cm high door centred at the bottom of one of the walls. A DLP projector (Optoma GT760) was used to deliver visual stimuli and illuminated a screen above the arena with a grey background (mean luminance: 80 candela/m^2). The arena was also illuminated by infrared LED strips positioned above it for tracking. Mouse movements were recorded with a near infrared camera (Blackfly S, BFS-U3-13Y3M-C, sampling rate: 60Hz) fitted with a wide-angle lens (LM4NCL) and positioned over the arena. Frames were acquired continuously in Bonsai and temporally aligned with the visual stimulus by simultaneously acquiring the signal from an infrared LED turned on at the stimulus onset and positioned outside the arena.

Visual stimuli generated through BonVision were rear projected onto the screen positioned above the arena (distance to the arena floor: 35cm). The stimulus consisted a 2.5cm black dot that appeared on one side of the arena and then translated smoothly to the opposite side over 3.3s.

Each session started with an animal being placed into the centre of the arena and being left to explore the environment for 5-10 minutes. After exploration, visual stimuli were manually triggered once the mouse was in the centre of the arena. Animals were left in the environment for 5 minutes after the last stimulus was presented. All animals experienced two sessions per day for 5 consecutive days. On each day, consecutive sessions for the same animal were separated by at least one hour. At the end of each session, animals were moved back to the home cage.

We tracked mouse movement using Bonsai nodes to measure the mouse position during the entire

session. Post hoc behavioural analysis was carried out using custom-made MATLAB scripts. We identified periods of freezing or flight responses based on the animals' speed: freezing was defined as a drop in the animal speed below 2cm/s that lasted more than 0.1s, and flight responses were defined as an increase in the animal running speed above 40cm/s. Responses were only considered if they occurred within 3.5s from stimulus onset.

Surgery

Mice were implanted with a custom-built stainless steel metal plate on the skull under isoflurane anaesthesia. The area above the left visual cortex was kept accessible for electrophysiological recordings. A ~1mm craniotomy was performed either over the primary visual cortex (2mm lateral and 0.5mm anterior from lambda) or superior colliculus (0.5mm lateral and 0.2mm anterior from lambda). Mice were allowed to recover for 4-24 hours before the first recording session.

We used a virtual reality apparatus similar to those used in previous studies (Schmidt-Hieber & Hausser, 2013; Muzzu, Mitolo, Gava & Schultz, 2018). Briefly, mice were head-fixed above a polystyrene wheel with a radius of 10cm. Mice were positioned in the geometric centre of a truncated spherical screen onto which we projected the visual stimulus. The visual stimulus was centred at +60° azimuth and +30° elevation and had a span of 120° azimuth and 120° elevation.

Neural Recordings

To record neural activity, we used a multi electrode array probes with two shanks and 32 channels (ASSY-37 E-1, Cambridge Neurotech Ltd., Cambridge, UK). Electrophysiology data was acquired with an OpenEphys acquisition board connected to a different computer from that used to generate the visual stimulus.

The electrophysiological data from each session was processed using Kilosort 1 (Pachitariu, Steinmetz, Kadir, Carandini & Harris, 2016). We synchronised spike times with behavioural data by aligning the signal of a photodiode that detected the visual stimuli transitions (PDA25K2, Thorlabs, Inc., USA). We sampled the firing rate at 60Hz and then smoothed it with a 300ms Gaussian filter. We calculated receptive fields as the average firing rate or local field potential elicited by the appearance of a stimulus in each location (custom routines in MATLAB).

Methods References

Muzzu, T., Mitolo, S., Gava, G. P., & Schultz, S. R. (2018). Encoding of locomotion kinematics in the mouse cerebellum. *PLoS ONE*, *13*(9). https://doi.org/10.1371/journal.pone.0203900

Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M., & Harris, K. (2016). *Fast and accurate spike sorting of high-channel count probes with KiloSort*.

Schmidt-Hieber, C., & Hausser, M. (2013). Cellular mechanisms of spatial navigation in the medial entorhinal cortex. *Nat Neurosci*, *16*(3), 325–331.