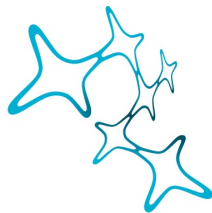

Design of Virtual Reality Systems for Animal Behavior Research

Nicholas Andrew Del Grosso



Graduate School of
Systemic Neurosciences
LMU Munich



*Dissertation der Graduate School of Systemic Neurosciences der
Ludwig-Maximilians-Universität München*

August, 2018

Supervisor:
Prof. Dr. Anton Sirota
Chair for Cognition and Neural Plasticity
Ludwig-Maximilians-Universität München

First Reviewer: Prof. Dr. Anton Sirota
Second Reviewer: Prof. Dr. Stefan Glasauer
External Reviewer: Dr. Stephan de la Rosa

Date of Submission: August 7, 2018
Date of Defense: November 30, 2018

These miraculous machines!
Do we shape them
Or do they shape us?
Or reshape us from our decent, far designs?

But we are learning.
We are learning to build for the future
From the ground up.

The Ground from Under Your Feet by Russell Lord
The Saturday Review, August 7, 1948

Abstract

Virtual reality (VR) experimental behavior setups enable cognitive neuroscientists to study the integration of visual depth cues and self-motion cues into a single percept of three-dimensional space. Rodents can navigate a virtual environment by running on a spherical treadmill, but simulating locomotion in this way can both bias and suppress the frequency of their behaviors as well as introduce vestibulo-motor and vestibulovisual sensory conflict during locomotion. Updating the virtual environment via the subject's own freely-moving head movements solves both the naturalistic behavior bias and vestibular conflict issues. In this thesis, I review elements of self-motion and 3D scene perception that contribute to a sense of immersion in virtual environments and suggest a freely-moving CAVE system as a VR solution for low-artifact neuroscience experiments. The manuscripts describing the 3D graphics Python package and the virtual reality setup are included. In this freely-moving CAVE VR setup, freely-moving rats demonstrate immersion in virtual environments by displaying height aversion to virtual cliffs, exploration preference of virtual objects, and spontaneously modify their locomotion trajectories near virtual walls. These experiments help bridge the classic behavior and virtual reality literature by showing that rats display similar behaviors to virtual environment features without training.

Contents

Abstract	v
1 Introduction	1
1.1 Applications of Virtual Reality in Neuroscience Research	1
1.2 Immersing Animal Subjects with Wide-Field Displays	3
1.3 Providing a Sense of Space by Rendering Realistic Virtual Environments	5
1.4 Simulating Locomotion through a Virtual Environment	6
1.5 Display Lag Requirements for Self-Motion-Induced Vection	9
1.6 Are Rodents Immersed in Virtual Environments?	11
1.7 Implementing a Freely-Moving CAVE System for Neuroscience Re- search in Rodents	13
1.8 Description of Manuscripts	14
2 Ratcave: A 3D Graphics Python Package for Cognitive Psychology Experi- ments	15
2.1 Author Contributions	16
2.2 Abstract	17
2.3 Introduction	18
2.4 Software Description	19
2.4.1 Built-In Primitives and Graphics Resources	19
2.4.2 Rendering 3D Meshes in Ratcave	19
2.4.3 Manipulating Perspective in 3D Scenes: Ratcave’s Camera Class	20
2.4.4 Working with Groups of Meshes: Scenes and Scene Graphs . .	20
2.4.5 Integrating Multiple Rotation Coordinate Systems	21
2.4.6 Manipulating Data on the Graphics Card	22
Vertex Arrays	22
Uniform Data	23
Texture Data	24
2.4.7 Writeable Textures: Building Deferred Rendering Pipelines. . .	25
2.4.8 OpenGL Shader Programs	26
2.5 Discussion	29
2.5.1 Advantages of Targeted Software in the Open Source Commu- nity	29
Commercial Game Engines in Place of Scientific Stimulus Pack- ages	30

3	A Virtual Reality System for Freely-Moving Rodents	33
3.1	Author Contributions	34
3.2	Introduction	35
3.3	Results	38
3.3.1	Ratcave VR: VR system for freely moving rodents	38
3.3.2	Flexible design, calibration and mobility of the VR arena	38
3.3.3	Low latency motor-visual feedback of the Ratcave VR system	39
3.3.4	Visual cues enhancing VR immersion	39
3.3.5	Testing spontaneous behavior of rats in the Ratcave VR	40
3.3.6	Virtual cliff avoidance experiment	40
3.3.7	Interaction with virtual walls	40
3.3.8	Exploration of virtual objects	42
3.3.9	Effect of virtual environment on hippocampal spatial map	42
3.4	Discussion	47
3.5	Online Methods	50
3.5.1	Ratcave VR	50
	Hardware Setup	50
	Software	50
	Code Availability	51
	Data availability	51
	VR implementation	51
	Latency measurement	53
3.5.2	Animal Experiments Methods	54
	Subjects	54
3.5.3	Behavioral experiments	54
	Virtual cliff experiment	56
	Virtual wall experiment	57
	Virtual object exploration	59
	Virtual spatial novelty detection experiment	60
	Virtual environment shift experiment	63
	Surgery and electrophysiological recordings	63
	Data acquisition and processing	63
3.5.4	Data Analysis	64
	Data representation and statistics	64
	Behavioral state classification	65
	Electrophysiology Data Analysis	65
	Brain state segmentation	65
	Place cells analysis	65
4	Discussion	67
4.1	Development of the ratCAVE VR Setup	67
4.1.1	The 3D Graphics Software	68

4.1.2	The Arena	68
4.1.3	The Head Mount	69
4.1.4	The Projector Calibration Method	70
4.1.5	The Arena Mesh Reconstruction Method	70
4.2	Increasing VR Immersion: Further Directions	72
Bibliography		75

List of Figures

1.1	Projection Surface Geometries used in Rodent VR Setups.	4
2.1	3D Mesh primitives supplied by Ratcave	19
2.2	Creating a 3D Mesh in Ratcave	19
2.3	3D Lighting Effects in Ratcave	19
2.4	Positioning, Rotating, and Scaling a Mesh	20
2.5	Ratcave’s compatibility with other Python graphics packages	20
2.6	Ratcave’s Camera class	21
2.7	Ratcave’s Scene class	21
2.8	Defining Spatial Relationships via a Scene Graph	21
2.9	Warping Meshes in real-time	23
2.10	Sending uniform data to the OpenGL Shader program	24
2.11	Legacy vs Modern OpenGL interfaces	24
2.12	Applying images to Meshes via Textures	25
2.13	Two-Pass Rendering using Framebuffers	26
2.14	A GLSL Vertex Shader	26
2.15	A GLSL Fragment Shader	27
2.16	Ratcave’s Shader class	27
2.17	Project codebase sizes among psychology stimulus libraries	30
2.18	Class dependency tree of Python stimulus presentation packages.	30
3.1	Schematic of the Ratcave VR VR components.	39
3.2	Virtual Cliff Avoidance Experiment	41
3.3	Virtual Wall Exploration Experiment	43
3.4	Virtual Object Exploration Experiment	44
3.5	Impact of virtual environment on hippocampal spatial representation	46
3.6	Photographs of the Ratcave VR setup	50
3.7	Ratcave VR hardware-software components flowchart	52
3.8	Cube-mapping and image rendering.	53
3.9	Motion-to-photon lag measurement of Ratcave VR system.	55
3.10	Spatial Novelty Exploration Experiment	62
3.11	Histological stained slice verification of electrode implantation location in hippocampus	64
4.1	Development of the ratCAVE 3D graphics engine.	68

4.2	The ratCAVE arena design.	69
4.3	Development of the Rodent Head Mount.	69
4.4	Development of the Projector Calibration Method	70
4.5	The Arena Mesh Reconstruction Algorithm.	71

*Dedicated to Prof. Dr. Josephine Wilson,
who gave me the tools I needed to venture out into the world
unafraid.*

Chapter 1

Introduction

1.1 Applications of Virtual Reality in Neuroscience Research

The rat seeks out his target: a white cylinder hovering impossibly in the air, hidden in a forest of black flying cylinders in his computer-generated world. Found, he dashes forward, his paws forcing the ground backward even as his body stays in place. His pixelated world streams past him until, several meters later, he reaches his goal and a spurt of sweetened milk is delivered to a tube by his mouth, a reward for a task well-done. The world fades away into the darkness, and when the light lifts again a second later a totally new virtual environment is revealed, a new forest of floating cylinders, ready to explore.

This search task, successfully performed by rats in Hölscher et al. 2005, announced to the neuroscience research community that virtual reality had finally arrived to animal research, and with it all the experimental possibilities the technology enabled. With virtual reality, researchers could more fully control the visual components of a subject's three-dimensional environment with high spatiotemporal precision while the subject locomoted about a space, a benefit previously restricted to head-fixed behavioral setups. Besides manipulating a specific stimulus' properties, experimenters could use virtual reality to fully isolate and manipulate these visual environments at will, enabling sophisticated experimental paradigms with randomized arena layouts (Aronov and Tank, 2014), objects that appear and disappear over time (Harvey, Coen, and Tank, 2012; Pinto et al., 2018), and complex and dynamic labyrinths (Thurley et al., 2014). Because all elements of the virtual environment are simulated, experimenters can also use virtual reality to study a single sensory modality without simultaneously introducing potentially-confounding, olfactory and sensorimotor cues normally present during object and arena exploration. Finally, all of the experimental manipulations in virtual reality are fully automatic, without any need for the experimenter to manipulate the stimuli themselves (e.g. to replace, clean, or reposition objects in an arena, potentially introducing new confounding interactions with their subjects) or even physically enter the recording chamber during a session.

More interesting, perhaps, is that virtual reality enables researchers to experiment with space itself, building infinitely-large arenas, topologically-impossible spaces, or abstract visual stimuli (e.g. Gabor patches, edgeless objects) into a rich, interactive 3D space. As a result, virtual reality setups allow researchers to manipulate aspects of the sensorimotor loop, helping them build experimental models of self-motion perception, spatial navigation strategies, and multisensory integration in ways not possible in normal behavioral setups. Finally, virtual reality can be used to give heterospecific species the appearance of conspecifics, enabling them to interact in real-time in ways not attainable in normal biological settings (Normand et al., 2012) to study or take advantage of social learning to reduce task training time, as has been demonstrated through robot-rodent social learning paradigms (Gianelli, Harland, and Fellous, 2018).

Virtual reality is even useful for researchers who don't plan to build complex 3D environments. By explicitly designing the environment and recording each moment of interaction with it, more detailed behavioral data is available for researchers to analyze and relate to the environmental context. The ability to simulate locomotion in a head-fixed subject also enables more precise neurophysiological recordings, including high-resolution calcium imaging. By relating this imaging data with the rodent's locomotion in virtual environments, major breakthroughs have been achieved such as the discovery of a secondary interneuronal system that selectively encodes spatial location during immobility from dendritic spike population data (Sheffield and Dombeck, 2015a) and long-term place field plasticity (Kay et al., 2016; Arriaga and Han, 2017).

Since its introduction to the rodent neuroscience community in 2005 (Hölscher et al., 2005), a wide variety of VR systems have been developed in animal behavioral laboratories, with the design of each system guided by the research group's scientific interests and the limitations of their workspaces and budgets (for reviews of VR use in neuroscience, see (Tarr and Warren, 2002; Bohil, Alicea, and Biocca, 2011; Dombeck and Reiser, 2012; Stowers et al., 2014; Thurley and Ayaz, 2016; Cullen and Taube, 2017). Building a virtual reality setup for research is, at its heart, quite simple: a 3D graphics system displaying a virtual environment around the subject is connected to a motion tracking system in a closed loop, thereby allowing interaction with that environment. However, the parameters that qualify a system as being "virtual reality" setups are far from unified, and because the VR system in every laboratory is different, the interpretations of animal behavior and neurophysiological measurements of animal virtual environment interaction during VR experiments should also be different for each group. The selection and design of each part, therefore, requires careful consideration of the biophysics, neurophysiology, and cognitive attributes of the animal model to be immersed in the environment.

In this thesis, I highlight how elements of a virtual reality system influence the perception and cognitive processing of subjects, particularly emphasizing rodent animal models. These principles were used to guide the development of the ratCAVE virtual reality experimental setup and software in the laboratory of Prof. Dr. Anton Sirota during my doctoral candidate research studies, described in the attached manuscripts. By approaching VR system design from a foundation of perceptual principles, novel solutions were found that enable more immersive virtual environments with fewer sensorimotor conflicts, higher performance characteristics, and greater experimental flexibility than other systems in the animal VR field. Behavioral experiments conducted in this VR setup have allowed us to demonstrate that rodents intuitively understand, without training, virtual environments in this setup to be three-dimensional scenes and interact to them in a consistent manner as directly compared to their real-world counterparts. In doing so, this project attempts to find bridges between the experimental literature in spatial navigation in virtual reality and freely-moving experimental setups.

1.2 Immersing Animal Subjects with Wide-Field Displays

When a subject is immersed in a virtual environment, they feel present and engaged in that environment. The sense of self-movement through a virtual environment helps increase immersion. Vection, the sense of self-movement generated purely visually from optic flow information, is especially acute during low-acceleration movements, for which the visual system is sensitive but the vestibular system is insensitive (McCrea et al., 1999). On the physiological side, low-speed rotational optic flow has been shown to influence spatial perception by temporarily shifting the compass-like direction selectivity tuning of head direction cells in the anterior dorsal nucleus (ADN) of the thalamus, indicating a general sense of self-rotation in space in spatial navigation systems in the brain (Knierim, Kudrimoti, and McNaughton, 1998; Arleo et al., 2013). Since self-motion-related optic flow is a full-field visual stimulus, it is highly-salient even to insects, who use optic flow cues to control flight altitude and distance from obstacles and goals (Straw, Lee, and Dickinson, 2010; Dittmar et al., 2010; Baird et al., 2013).

Because the induced sense of vection from optic flow is stronger the greater the visual angle that optic flow subtends, a VR display must naturally also be as wide-field as possible. Ideally, the display would encompass all of the subject's field of view to increase the animal's sense of immersion, either by being very large or very close to the subject's eyes (e.g., head-mounted smartphone screens viewed through fresnel lenses, as used in many human VR systems). For rodent subjects, a typical solution has been to use curved mirrors to project onto large, curved projection displays whose image covers as much of the rodent's visual field as possible. The exact shape of these displays varies between groups; they can be spherical (Lee et al., 2007) (Figure 1.1a), cylindrical (Fry et al., 2004; Aronov and Tank, 2014; Driscoll

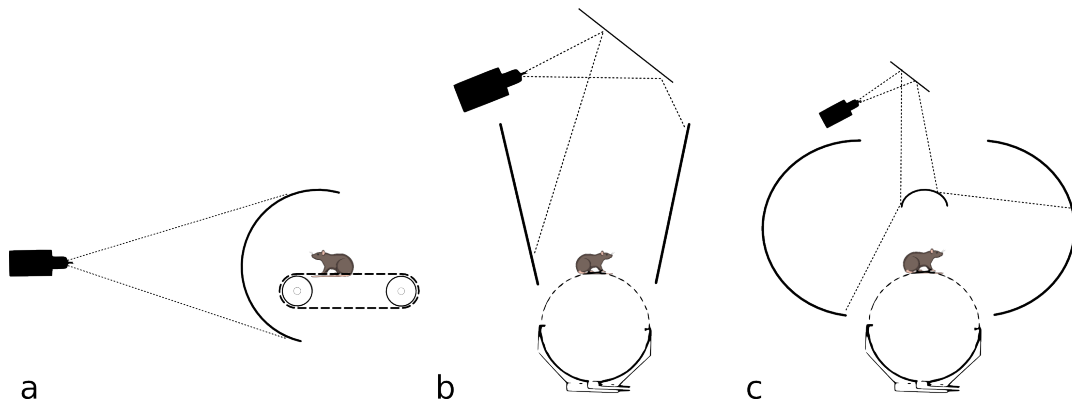


Figure 1.1: Examples of Projection Surface Geometries used in Rodent VR Setups. a) A half-spherical screen, as used in Lee et al. 2007. b) a cylindrical screen, as used in Aronov and Tank 2014. c) a toroidal screen, as used in Hölscher et al. 2005.

et al., 2017) (Figure 1.1b), or toroidally-shaped (Hölscher et al. 2005; Dombek and Reiser 2012, Figure 1.1c), or any other geometrical arrangement that is either front- or back-projected onto. Each of these geometries has its advantages: toroids maintain fixed screen-to-rodent distance and cover the entire visual field, half-spheres can be back-projected from a single video projector, and cylinders are simpler to design and install.

In order to render a 3D environment onto a curved projection screen, a virtual reality system must take into consideration the relative position of the video projector and surface (“projection mapping”) as well as the relationship between the projection surface and the subject (“perspective correction”); errors in these calculations result in both static geometric distortions in the virtual environment projection and nonlinearities in the optic flow field during locomotion. Projection mapping is done using a screen-specific solution, either using a parametric equation to map the flat projected image to the screen, as is done with toroidal screens (Aronov and Tank, 2014), or it is simply skipped by using multiple projectors, each projecting onto a flat screen, as done in the first Computer-Assisted Virtual Environment (CAVE) system, the base VR model from which all rodent VR setups derive their design. CAVE, developed by Carolina Cruz-Neira (Cruz-Neira et al., 1992), involves projecting the virtual environment onto the floors and walls of the subject’s arena from 6-12 video projectors. Because rodents are small and often position-fixed on a treadmill, a single projector and more complex projection mapping allow researchers to simplify their setups, reducing the need for the graphics computer cluster and large space requirements required in the original CAVE systems. While the original CAVE performed perspective-correction on a per-display basis, more recent graphics libraries allow for an approach called “cube mapping”, which more efficiently maps a virtual space onto multiple 2D projections of that space from the subject’s perspective. When used together, cube mapping and parametric projection mapping techniques allow a geometrically-correct virtual environment to be projected from a rodent’s

viewpoint.

It should be mentioned that not all rodent VR systems use video projectors to display the virtual environment; some laboratories make use of flat-screen monitors arranged in concave shapes about the subject (5 monitors: Radvansky and Dombeck 2018; Heys, Rangarajan, and Dombeck 2014; 3 monitors: Ayaz et al. 2013; 2 monitors: Keller, Bonhoeffer, and Hübener 2012; 1 curved monitor: Sheffield and Dombeck 2015b; Arriaga and Han 2017). These systems are simpler to install and have much higher display resolutions than the single-projector setups described above, but often sacrifice some amount ofvection by covering a smaller field of view (e.g. Heys, Rangarajan, and Dombeck 2014: 67 degrees vFOV). A great use case for a multi-monitor system is from the Prusky lab, which renders perspective-corrected rotating cylinders from the perspective of a small central platform in order to measure visual acuity in rodents (Douglas et al., 2005).

While the display technologies employed by each VR setup may differ, no particular display technology has a clear advantage over another. The design of a VR setup should thus be chosen carefully based on the parameters of the researcher's intended experiments and the sensory system of the animal model. Video projectors, for example, tend to have poor brightness and contrast specifications, but rodents' high sensitivity to light makes this an advantage; in fact, often researchers decrease the maximum projection brightness levels of their displays to comfortable ranges using lower-power lamps or neutral density filters (as seen in Arriaga and Han 2017). The high resolution of multi-monitor systems makes them a good fit for visual acuity measurements, but may not be necessary for many VR applications as the resolution of these monitors exceeds the visual acuity of mice and rats (0.5-1.5 cycle per degree, Burn 2008). Great examples of species-specific displays can be found in the fly literature, where VR setups with custom LED matrix displays have been built that sacrifice display resolution in order to gain high visual contrast and high display frequencies (Strauss, Schuster, and Götz, 1997; Reiser and Dickinson, 2008).

1.3 Providing a Sense of Space by Rendering Realistic Virtual Environments

While dynamic visual cues (e.g., optic flow and motion parallax) help deliver the impression of a 3D scene, additional immersive visual cues can be added that provide further information about the scene's geometry and the subject's position relative to virtual objects in the form of 3D lighting cues. Lighting cues provide a wealth of information: diffuse reflections reveal the relative orientation of an object to a light source independent of the subject's position, specular reflections modulate diffuse reflections by the relative position of the viewer, ambient light gives a sense of the environment's size, and shadows show the relative distances between objects and nearby surfaces. Simulating the contribution of these cues to render 3D graphics is

commonly done using the Phong reflection algorithm and shadow projection mapping (Phong 1975; for a review of algorithms related to shading, see Schlick 1994). Humans use these lighting cues to improve their performance in estimating the location of 3D objects in a rendered scene (Westerman and Cribbin, 1998; Hubona et al., 1999; Schrater, 2000).

Although these lighting algorithms have been in use for decades, thus far they are rarely employed in the virtual environments used in neuroscience research. This could be because the underlying graphics engines they rely on don't support those lighting features; the Quake 2 engine, for example, can only perform pre-baked diffuse lighting calculations that are challenging to implement and yet is used by several VR setups (Harvey et al., 2009; Heys, Rangarajan, and Dombeck, 2014; Sheffield and Dombeck, 2015b). Other VR projects directly write the OpenGL graphics code themselves, which provides new opportunities for these lighting calculations but requires a new graphics engine to be created, a time-intensive task that often produces lab- and hardware-specific software and have thus far not demonstrated the use of the lighting models described (Unnamed Matlab software: Hölscher et al. 2005; "TrackFly": Fry et al. 2008; "CalVR": Schulze et al. 2013). This may be starting to change; rich, 3D-lit environments can be seen in a recent publication by the Bartos group where the environment was rendered with the open-source Blender3D software (Hainmueller and Bartos, 2018).

1.4 Simulating Locomotion through a Virtual Environment

The key difference between a virtual environment and a simple 3D rendering is that the virtual environment is interactive; the viewer can move through the environment and affect change on the virtual objects it contains at will. The more natural this interaction seems, the more immersed the viewer feels in the virtual environment. These interactions are species-specific; for example, simulating flight for moths (the first VR spatial navigation system for animal research) is done by body-fixing the moth, simulating wind, and measuring the tilt of its thorax flight simulation system to allow the moth to turn (Gray, Pawlowski, and Willis, 2002).

Rodent locomotion through a virtual environment is most commonly simulated by mounting the subject on top of a passive spherical treadmill and measuring the treadmill's movement (for a comprehensive review of rodent VR setups, see Thurlley and Ayaz 2016). The treadmill approach, developed first for insects using table-tennis balls mounted on ball bearings (the "Orientometer": Kerfoot 1968) and later a cushion of air (Carrel, 1972), requires a ball whose arc approximates a surface flat enough to be walked on comfortably and a rotational moment of inertia matching the weight of the animal mounted on it, a design that works for mice with hollowed-out styrofoam balls and rats with specially-designed air cushions to reduce the noise

levels created by the air current suspending the treadmill to acceptable levels and enable auditory stimulus presentation for rats (Aharoni, 2013). Often, the subjects are head-fixed in place above the treadmill, which besides keeping the subject centered, enables stable neurophysiological measurements while the subject runs and simplifies positioning of the behavior rewarding apparatus. In addition, the treadmill can constantly move in any direction, enabling infinitely-large virtual environments for subjects to navigate, a major advantage considering the space limitations of a typical laboratory setting. Using a treadmill is also a fairly versatile strategy, as similar systems have been developed for research with flies (Jung, Borst, and Haag, 2011), fish (Ahrens et al., 2012), and the praying mantis (Nityananda et al., 2016).

Actively running on a treadmill does not necessarily mean that the subject perceives themselves to be moving through space, however; if the interactions with the setup contain low “motor affordance”, the expected amount of interaction that the subject perceives, the simulation will be unconvincing. This can occur, for example, if the treadmill is uncomfortable to run on or requires unnatural movements to operate, as seen in the steering-like motion required for turning in head-fixed setups, unnaturally-curved 2D locomotion trajectories, a reduced frequency of trajectory direction changes, and slower locomotion, as compared to real-world locomotion (Chen et al., 2018). The more complex the interaction one wants to simulate, the more challenging it is to meet expected affordance; lowering this expectation can be helpful, for example, with narrow virtual tracks, where locomotion more closely matches that of linear treadmills (Sheffield and Dombeck, 2015b; Tennant et al., 2018). To better-simulate motion in two dimensions, some research groups use a freely-rotating, body-fixed treadmill setup instead (Thurley and Ayaz, 2016). While this has resulted in some immediate improvements, it does not necessarily completely recover perceived motor affordance. Tight restraint, for example, similar to the tight vests used in body-fixed setups, has been shown to reduce head direction-specific and anticipatory firing in the anterior dorsal nucleus of the thalamus (ADN), an effect that disappears when rats are only loosely restrained (Foster, Castro, and Bruce, 1989; Bassett et al., 2005). While this was demonstrated in passively-moved subjects, it has also been shown with active movements by demonstrating that flies require free head movements in order to respond to closed-loop visual flight control, despite the fact that their flight patterns are unaffected by head restraint (Stowers et al., 2017). Perceived restraint levels can thereby reduce motor affordance, affecting the rodents’ sense of self-motion as a result.

Simulating locomotion with treadmills has another downside: because the subjects do not physically move in space, they do not receive locomotion-related translational vestibular sensation, resulting in both vestibulomotor and vestibulovisual conflict. While this may be a useful property for some specific experiments (Villette et al., 2015; Chen et al., 2018, e.g), removing vestibular inputs during active motion produces over-inhibition by the motor system onto the visual system (Becker

et al., 2002; Jürgens and Becker, 2006; Frissen et al., 2011; Telford, Howard, and Ohmi, 1995), an efferent copy signal normally used for encoding current movements and anticipating future head direction state (Meer et al., 2007; Carriot, Brooks, and Cullen, 2013). Vestibular information is an important part of the self-motion perceptual system, converging with visual, proprioceptive, and motor information as early in the brain as the second synapse and in the vestibular nucleus (reviewed in Cullen and Taube 2017). As seen by observing motor efferent copy-related inhibition from the motor system during active head movements (McCrea et al., 1999; Roy and Cullen, 2001; Cullen and Roy, 2004), vestibular sensation embeds itself inextricably from multisensory information flow in order to more accurately estimate self-motion (Stackman, Clark, and Taube, 2002). Head velocity encoding may also have an important role in position and direction coding, as seen in measurements of the rat dorsal tegmental nucleus (DTN), which contains head velocity cells that encode head movement direction and speed of up to 200 degrees per second and directly excites ADN head direction cells, with DTN lesions reducing ADN head direction cell observations (Blair and Sharp, 1995; Bassett and Taube, 2001; Bassett, Tullman, and Taube, 2007), and conversely disrupt the directional tuning of head direction cells when vestibular sensation is altered, whether by occluding the vestibular canal (Muir et al., 2009), lesioning the vestibular nerve (Stackman and Taube, 1997), or using mutant otolith particle-missing mice (Yoder and Taube, 2009). It is, of course, possible to perform spatial tasks without vestibular cues (Gibson, Butler, and Taube, 2013), but because disrupting spatially-tuned cell firing produces undesired physiological artifacts, maintaining accurate vestibular input during self-motion simulation should be considered an essential factor for simulating self-motion in virtual reality setups.

How can VR setups provide sensory vestibular input while still keeping a rodent fixed in place on a treadmill? Switching from head- to body-fixation is a good first step, as it allows the rodent to freely move their head to explore the virtual space and thereby sense rotational vestibular inputs. At the same time, however, it introduces new vestibulovisual conflicts, as high-speed translational vestibular inputs that occur during head movements remain in conflict with movements in the virtual environment. One compromise solution is to keep the head fixed in position and restrict head rotations to only the vertical axis, a strategy that has the further benefit of maintaining an imaging focal plane relative to a fixed microscope (Chen et al., 2018). To deliver translational vestibular stimulation, VR setups used in vision-vestibular research and driving simulators deliver vestibular sensation with a moving platform or robotic arm, moving the whole setup in order to use the force of gravity to simulate movement acceleration (Wiener, Berthoz, and Zugaro, 2002), but in these cases the subject is sitting still; pairing these setups with a passive treadmill may prove challenging to implement.

Rather than simulating movement, some VR setups allow full freedom of movement in space through head tracking. This can be found both in head-mounted VR setups and in freely-moving CAVE (“Computer-Assisted Virtual Environment”) setups, where the subject’s head position is tracked continuously via accelerometers, gyroscopes, or multi-camera arrays. The display is back-projected onto the walls and floor of a room, and the subject’s head movements are tracked with an external camera array and used to continuously update and reproject the virtual scene onto the display from the subject’s viewpoint. A head-tracking approach has some advantages compared to treadmill systems. First, because the head is physically moving through space, full motor affordance is maintained, including more complex behavioral interactions like rearing and leaping. Second, vestibulomotor coherence is adequately maintained, since the animal does physically move in space. Finally, CAVE these systems are potentially scalable across species, as they only require the head position of the subject as an input. Freely-moving CAVE setups, first demonstrated for animal research with successful closed-loop visual stimulation in flies (freely-walking: Strauss, Schuster, and Götz 1997; freely-flying: Straw, Lee, and Dickinson 2010), can be scaled up to rodent arena-sized setups with the same benefits. By maintaining a consistent relationship between the animal’s movement and the visual virtual environment, CAVE setups can provide a richer spatial landscape than body-fixed VR setups without the engineering complexity of simulating all aspects of the rodent’s interactions with an environment. Because functional brain recordings can be performed in freely-moving animals using chronically-implanted multicellular recordings with tetrodes or silicon probes (Chen et al., 2013), optic fibers, lenses, and head-mounted microscopes for brain imaging (Silva, 2017), and even in-vivo juxtacellular or intracellular recordings in freely-moving animals (Harvey et al., 2009; Lasztóczy and Klausberger, 2014), the simplicity of CAVE setups makes them an ideal candidate experimental setup for adding virtual reality to the range of neuroscience tools.

1.5 Display Lag Requirements for Self-Motion-Induced Vection

Because simulating a coupling between behaviors in the real world and their consequences in a virtual world requires computational time, artifactual vection from visuomotor mismatch can be accidentally induced by a slow VR system when the subject changes behavior more quickly than the system can update. This latency between the motion-stimulus coupling, dubbed “display lag”, induces vection both at the start of change in behavior (no visual response until the system catches up) and afterward (an inappropriate visual response to a behavior that has already passed), resulting in a highly-nonlinear visual input that can disrupt interpretation of physiological responses to virtual motion, produce a wobbling visual sensation called “oscillopsia”, or even produce motion sickness in the subject (dubbed “VR sickness”

or “simulation sickness” in the virtual reality literature) if the display lag is too high (Allison et al., 2001). Because motor actions modulate sensory information processing throughout the brain and change sensory processing itself (Stringer et al., 2018); for a comprehensive review, see Crapse and Sommer 2008), sensorimotor mismatch can also introduce artifacts in physiological studies. Neurons involved in low-level, early visual processing increase their firing rate during locomotion in both flies and mice (Jung, Borst, and Haag, 2011; Ayaz et al., 2013), a change that increases the size of visual fields and reduces the cell’s surround inhibition inputs (Niell and Stryker, 2010). Increased visual motion sensitivity during treadmill walking vs immobility has also been demonstrated in head-fixed flies (Chiappe et al., 2010). Whether from a perceptual or physiological perspective, minimizing display lag should be therefore be a high-priority goal when designing a VR setup.

Since display lag cannot be completely eliminated, some minimal display lag performance level guidelines for rodent behavioral and physiological experiments would be helpful for researchers. Some attempts have been made at estimating the lower bounds of display lag sensitivity in humans, but the systems used in these studies themselves have had latencies higher than the lower bounds of their subjects’ perception (Ash et al., 2011). In any case, a single value is difficult to pin-down; lag detection is movement-specific, with the quickest movements producing the greatest visual offsets, and different VR setups restrict movement in different ways. The value is also nonstationary, since humans are capable to some degree of adapting to different display lag levels over time (Draper, 1998). The connection between self-motion cues and visual processing seems to maintain its plasticity in adulthood, shown by the mouse visual cortex adapting to reversed rotational optic flow over time in a head-fixed VR setup (Leinweber et al., 2017), a sensorimotor integration model that is also being explored for whisker sensation and movement in barrel cortex in virtual reality (Sofroniew et al., 2014; Sofroniew et al., 2015). Augmented-reality setups like in CAVE systems (both head-fixed and freely-moving) additionally increase display lag sensitivity by providing “visual slip” information—the difference between real-world and virtual-world cue movement dynamics—thereby requiring even lower display lags in order to surpass human sensitivity levels (Zheng et al., 2014; Jerald, 2004). Finally, the maximum display lag is experiment-specific, with physiological studies requiring lower latencies than behavioral studies in order to more-easily link neural activity to sensory information.

Most rodent VR papers only report an estimate based only on the framerate of their display, a technique that severely underestimates the total VR latency of a system (Aharoni, 2013). Rodent VR systems typically use 60-Hz displays, and the slowest system I found in my literature review had 250 msec of display lag (Lee et al., 2007). For comparison, the potentially-fastest system found in my literature review was used for flies, with a 400-Hz display update rate reported (Reiser and Dickinson, 2008). While some solutions for lag reduction require custom hardware, 120-Hz

consumer video projectors with latency-reduction features have been available for some time, and are a simple way to reduce display lag in existing setups. The topic of latency in virtual reality has become a major area of research in computer science, however, and some rendering methods have been reported that may enable latencies of less than a millisecond, as is being developed in augmented-reality displays to minimize visual slip (Jerald, 2004; Kijima, Eijiroh, and Ojika, 2001).

Freely-moving CAVE systems, though, already have attained zero display lag for rotational movements of the head, a consequence of their always-360-degree display strategy. This 360-degree display requirement has traditionally been a technically-limiting factor in CAVEs for human setups, requiring upwards of 12 video projectors simultaneously projecting onto screens placed around the subject and rendering both eyes' viewpoints, which comes with a high price tag and extra translational display lag associated with extra processing needed to drive these extra displays. Rodent experiments, on the other hand, often take place in relatively small arenas, and rodents' comparatively-low visual spatial acuity makes it possible to split a single video projector's image between all projection surfaces at once, both decreasing the system cost and the processing time required to render the virtual environment. While this does create a difference in display lag for different types of movement, it at least eliminates display lag-induced vection for the type of behavior most likely to induce it in the first place: high-speed head turns. As such, single-projector freely-moving CAVE systems are a good candidate for low-latency VR setups in rodent research.

1.6 Are Rodents Immersed in Virtual Environments?

Despite the technical challenges associated with producing a fully-immersive, perceptual artifact-free virtual reality experience, existing rodent VR systems have already demonstrated themselves to be sufficiently advanced enough for behavioral and physiological spatial navigation research. Rodents have been trained to locomote toward distal virtual objects in one- and two-dimensional environments and stop upon reaching them (Hölscher et al., 2005; Kaupert et al., 2017). They also learn to avoid virtual walls, which allows researchers to build virtual mazes, teach rodents to shuttle linear tracks (Harvey et al., 2009; Dombeck et al., 2010; Driscoll et al., 2017; Sato et al., 2017) and make decisions in Y-mazes (Thurley et al., 2014; Pinto et al., 2018). By combining these properties, further experiments in arena foraging (Chen et al., 2018) and paradigms like the Morris Water Maze have also been demonstrated to be viable (Aronov and Tank, 2014). Rodents may also have some natural understanding of the virtual environment before training, as seen by untrained mice avoiding parts of a circular maze overlooking a virtual cliff (Stowers et al., 2017), although the vast majority of rodent VR studies include a training element, making it difficult to separate spontaneous behaviors based on perceptual bias from trained behaviors. Zebrafish, for example, use optic flow to regulate their swimming speeds

(Ahrens et al., 2012; Jouary et al., 2016), and self-motion cues seem to be involved in training the visual system, as evidenced by the presence of optic-flow mismatch detection signals in visual cortex when the speed and direction of optic flow are selectively decoupled from locomotion (Keller, Bonhoeffer, and Hübener, 2012; Attinger, Wang, and Keller, 2017). Mice, however, are capable of using the visual cues supplied by the virtual environment to estimate their distance travelled, which may indicate an understanding of the 3D nature of the virtual scene (Tennant et al., 2018).

Physiological measurements in the hippocampus and entorhinal cortex also indicate self-movement perception in some rodent VR setups. Hippocampal place cells have been observed in mouse electrophysiological and two-photon imaging recordings in both linear track shuttling environments (Harvey et al., 2009) and two-dimensional foraging arenas, along with border cells and grid cells (Aronov and Tank, 2014; Heys, Rangarajan, and Dombeck, 2014). These spatially-modulated cell activity patterns are observed in both head-fixed and free-body-rotation VR setups, even displaying locomotion-direction specificity in linear track environments, although this finding may be visually induced (Harvey et al., 2009; Dombeck et al., 2010). Direct physiological comparisons between matching real-world and virtual environments, however, show a wide range of similarities and differences between the two setups; some studies find lower amounts of spatial information content encoded during locomotion in virtual environments than in real environments, including lower theta rhythm amplitude and larger grid field spacing (Ravassard et al., 2013; Chen et al., 2018), while others have found greater consistency in physiological parameters during real and virtual environment navigation (Domnisoru, Kinkhabwala, and Tank, 2013). Differences in behavior between real-world and virtual environments have also been found (Ravassard et al., 2013), although some of these differences may disappear with training (Chen et al., 2018). Finally, lesioning studies in MEC have shown that the same path integration systems relied on during real-world navigation are also required during virtual environment navigation, with MEC stellate cells needed for accurate running distance estimation (Tennant et al., 2018). These activity patterns may not be generated by self-motion perception, however; place-cell-like activity and increased theta rhythm is also generated through movement on a standing treadmill (Geiller et al., 2017). One potential test would be to look at place maps between “teleportation” trials, where a rodent is instantaneously moved to a different position in a virtual space, and looking for changes in the rodent’s internal map, as has been done in real-world “teleportation” studies using electronic and reward cues sets to instantaneously change the rodent’s spatial context (Jezek et al., 2011; Posani et al., 2017).

How can immersion be improved in VR setups? Increasing the sensory richness of the experience would certainly be helpful, as has been done by some groups (Auditory VR: Aharoni 2013; Olfactory VR: Radvansky and Dombeck 2018, Somatosensory VR: Sofroniew et al. 2014). Because many senses influence spatial processing

on a neural level (Goodridge et al., 1998; Save et al., 1998; Radvansky and Dombeck, 2018), increasing the overall amount of spatial information should also help; however, in practice it is disputed whether this has had any effect on the yield of physiological spatial navigation studies (for: Young, Fox, and Eichenbaum 1994; Villette et al. 2015; Geiller et al. 2017; against: Markus et al. 1994). Virtual reality simulations can also be improved by removing situations that break immersion; for example, collisions with virtual walls in passive treadmill setups prevent the subject from running through a wall in virtual space, despite the subject still being physically able to run forward on the treadmill, an immersion-breaking situation that has been addressed with some success by using active treadmills or touchable, robotic walls to establish a more consistent virtual experience (Kaupert et al., 2017; Sofroniew et al., 2014; Kaupert et al., 2017). Another potential immersion-breaking situation arises in experiments that randomize the subject's position between trials by passively "teleporting" them across the environment; because teleportation doesn't involve locomotion, visual movement cues, or vestibular sensation, the subject would need to reorient themselves every trial. Maintaining the path integration system with the virtual environment can be done by requiring the subject to actively move themselves to each new starting location. While training certainly helps the subject understand the nature of the simulation and form new links between sensation and movement, these situations do break the intuitiveness, and thus immersiveness, of the virtual environment.

Another underutilized technique is the use of "augmented reality" in experiments, which combine real-world features with 3D projections. These techniques have the potential to be more immersive than full VR systems, since they remove real-world stimulus conflict and have lower technology barriers to implementation. They are also valuable for observing complex, naturalistic behaviors. For example, two-dimensional moving black dot stimuli have been demonstrated to elicit attack behaviors in spiders and praying mantises, who seem to treat them as virtual prey (Fenk, Hoinkes, and Schmid, 2010; Nityananda et al., 2016). Virtual models of conspecifics can also elicit social interactions, such as swarming and chasing in zebrafish, a behavior that maintains some of the same social dynamics as in real-world social situations (Stowers et al., 2017; Nakayasu et al., 2017). As a form of augmented reality, freely-moving CAVE VR setups thereby allow the same level of behavioral complexity to be studied in rodents while maintaining a multisensory cohesion to produce a highly-immersive environment.

1.7 Implementing a Freely-Moving CAVE System for Neuroscience Research in Rodents

In order to design a virtual reality system that provides full vestibular, olfactory, and somatosensory self-motion cues to rats for spatial navigation and sensorimotor

integration research, I designed a freely-moving CAVE system that front-projects a virtual reality “layer” onto a more traditional arena used for rodent research. This system, dubbed “ratCAVE”, allows our lab to perform neurophysiology recordings without concern for self-motion simulation artifacts introduced by spherical treadmill systems. In order to reduce rendering latency artifacts and cross-visual cue conflict, the system was designed to have low-latency performance (less than 17-millisecond rendering latency), have high update frequencies (360 Hz tracking and displaying), enable vestibular stimulation (through arena movement and real-time geometry warping calculations) and accurate 3D spatial cues (modern 3D reflection algorithms, implemented in OpenGL).

The virtual-reality interface was simplified for neuroscience researchers by creating the software as a set of packages for Python, a popular open-source scripting language used in many research settings, including neuroscience. This approach provides flexibility and compatibility for many research groups by separating the tracking and graphics sides of VR software. For example, the ViRMeN Matlab OpenGL graphics package (Aronov and Tank, 2014) would be compatible with our “NatNet-Client” tracking library, and our Ratcave graphics could be combined with the “Flydra” tracking system (Straw, Lee, and Dickinson, 2010). Programming language-specific libraries are especially useful for researchers already using the language, as they allow the combination of the language’s ecosystem of tools to be easily merged into a given experiment.

1.8 Description of Manuscripts

This cumulative dissertation consists of two manuscripts, both intended for publication in peer-reviewed scientific journals. The first manuscript is a description of the Python 3D rendering package that made implementing a virtual reality system possible, and the second is a full description of the ratCAVE VR setup, including behavioral experiments that demonstrate the immersiveness of the ratCAVE virtual environments for untrained rats.

Chapter 2

Ratcave: A 3D Graphics Python Package for Cognitive Psychology Experiments

2.1 Author Contributions

Authors: Nicholas A. Del Grosso and Anton Sirota

N.A.D.G. designed and implemented the Ratcave graphics software, wrote the paper, and produced the figures and videos. A.S. discussed the software's development and commented on the manuscript.

2.2 Abstract

Few vision stimulus software packages and libraries offer support to cognitive scientists for the presentation of 3D stimuli. We present here a free, open source Python 3D graphics package called Ratcave that allows scientists to load, display, and transform 3D stimuli created in 3D modeling software. This package makes 3D programming intuitive to new users by providing 3D graphics engine concepts (Mesh, Scene, Light, and Camera classes) that can be manipulated using an interface similar to existing 2D stimulus packages. In addition, the use of modern OpenGL constructs by Ratcave helps scientists create fast, hardware-accelerated dynamic stimuli using the same intuitive high-level, lightweight interface. Because Ratcave supplements, rather than replaces, existing Python stimulus packages, scientists can continue to use their preferred libraries by simply adding Ratcave graphics to their existing experiments. We hope this tool will be useful both as a stimulus package and as an example of how tightly-focused libraries can add quality to the existing scientific open-source software ecosystem.

2.3 Introduction

Experimental paradigms employed in a wide class of disciplines, such as cognitive psychology or neuroscience, rely on software that presents stimuli to a subject, detects subject responses, and logs events for future analysis, all with high temporal accuracy. An ever-expanding list of other features included in this software are compatibility with third-party hardware devices (e.g. button boxes, amplifiers, eye tracking systems), support for custom experimental designs, and online analysis for adaptive stimulus sequences and selection and real-time manipulation of continuous stimuli (e.g. neurofeedback and virtual reality). A wide variety of closed-source commercial stimulus presentation software is available for researchers (e.g. Neurobs Presentation, BCI2000, SuperLab, E-Prime), but the Python programming language has a wide ecosystem of free, open-source alternative libraries that help scientists build their behavioral experiments (e.g. Psychtoolbox by Brainard 1997; PsychoPy by Peirce 2007; Vision Egg by Straw 2008; Expyriment by Krause and Lindemann 2014; for a review of psychophysics libraries, see Kötter 2009). However, researchers wishing to study 3D mental rotation, spatial navigation in 3D virtual environments, object recognition using 3D models, or even those who simply wish to present completely novel visual stimuli to their subjects are still unsupported by these libraries. Commercial 3D graphics software is available, but often comes with licensing costs (e.g. Unity3D, a popular software package, is free for hobbyists but requires an expensive license for university research groups) or has extensive learning curves. While 3D graphics options do exist in Python (e.g. Panda3D, PyOgre, Vizard), they are not cross-compatible with the other packages, thereby creating a compromise situation for researchers who need to mix the stimuli, devices, and logging features from one library with the 3D stimuli from other libraries.

In this paper, we present an open-source, cross-platform Python package called Ratcave that not only adds 3D graphics support to all Python stimulus packages, including VisionEgg, Psychopy, and Expyriment, it also provides a simple interface for building any visual stimuli and running them on a graphics card as OpenGL shader programs. In this paper, we review the core features of Ratcave and highlight key connections of its interface to underlying graphics programming strategies (a thorough manual, complete with API guide and tutorials for first-time users can be found at <https://ratcave.readthedocs.org>). By demonstrating the accessibility of 3D graphics programming through this new package, which derives its name from our high-speed RatcaveVR experimental setup (Del Grosso et al., 2017), we introduce this package as a useful addition to the existing ecosystem of psychology software for Python.

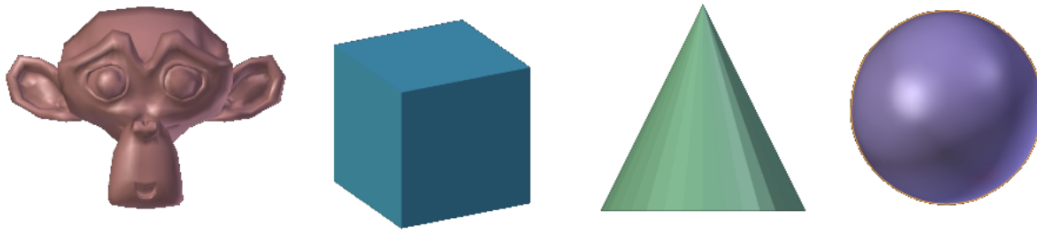


Figure 2.1: Some of the 3D Mesh primitives supplied with Ratcave.

```
import ratcave as rc

model_file = rc.resources.obj_primitives
cube = rc.WavefrontReader(model_file).get_mesh('Cube')
```

Figure 2.2: Importing Ratcave into the Python environment and creating a cube stimulus from the Ratcave's supplied primitive meshes.

2.4 Software Description

2.4.1 Built-In Primitives and Graphics Resources

In order to make 3D programming accessible, Ratcave comes with a collection of resources, including basic 3D object primitives (Figure 2.1 and 2.2) and a wide range of 3D lighting effects (Figure 2.3, Supplementary Video 1). This way, a user can get started quickly, writing customized code only when needed.

Creating Meshes, a term used for any 3D object in Ratcave, is done either by supplying the vertex coordinates as an array or by importing from a 3d-formatted file (for example, the popular Wavefront file format, for which Ratcave provides a parser). All objects in a Ratcave Scene (Meshes, Lights, and Cameras) can be repositioned, rotated, and scaled using an intuitive object-oriented interface (Figure 2.4).

2.4.2 Rendering 3D Meshes in Ratcave

Once a Mesh is loaded and positioned, it can be drawn in any active OpenGL window (e.g. a Psychopy window, Experiment window, Vision Egg window, etc) by

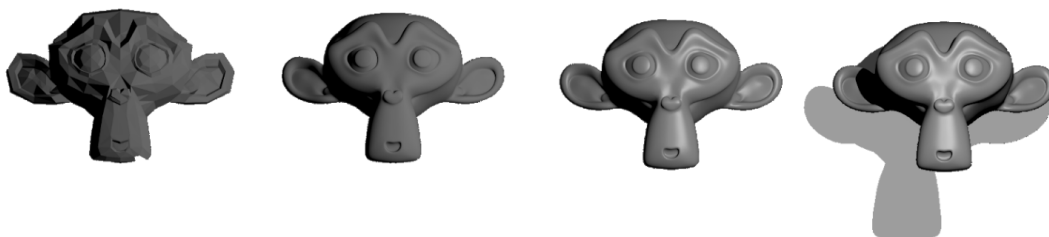


Figure 2.3: Examples of some simple 3D lighting effects available in Ratcave: diffuse and “glossy” specular reflections, ambient lighting, and shadows.

```
cube.position.x = 5
cube.rotation.xyz = 90, 0, 180
cube.scale.y = 0.5
```

Figure 2.4: Positioning, Rotating, and Scaling a Ratcave Mesh by assigning new values to their correspondingly-named attributes.

<pre> window = psychopy.visual.Window() while True: window.clear() with rc.default_shader: cube.rotation.y += 1. cube.draw() window.flip() </pre>	<pre> window = pyglet.window.Window() @window.event def on_draw(): with rc.default_shader: cube.rotation.y += 1. cube.draw() pyglet.app.run() </pre>
---	--

Figure 2.5: Demonstrating the flexibility of Ratcave by drawing a rotating cube in two different packages: Psychopy (left) and Pyglet (right). Although the syntax of each package differs from each other, the Ratcave drawing code is the same.

binding it to a Shader program using Python’s `with` keyword and calling its `draw()` method. Ratcave provides a default shader that performs many industry-standard 3D transformation and lighting steps (including diffuse and specular lighting, and shadow-mapping, Figure 2.3), allowing users to create useful 3D scenes from the beginning of their work with the package (Figure 2.5).

2.4.3 Manipulating Perspective in 3D Scenes: Ratcave’s Camera Class

Unlike in 2D graphics, where the screen’s pixels provide a natural coordinate space for positioning objects, a 3D scene is composed of 3D objects (“Meshes”) viewed from a given perspective (the “Camera”) which is projected down onto the 2D surface of the display. Positioning objects on-screen is further made intuitive by Ratcave’s Camera class, which functions similarly to virtual cameras in 3D modeling software. Besides being positioned and rotated to face an object, properties of the Camera’s intrinsic projection model (e.g. field of view, aspect ratio, and frustum cutoff thresholds, orthographic vs perspective projection) can be manipulated as well. To draw a Mesh from the perspective of the camera, it is bound by the user using a similar method as with the shader: using Python’s `with` keyword statement (Figure 2.6).

2.4.4 Working with Groups of Meshes: Scenes and Scene Graphs

Once all meshes are all loaded, they can be collected together in any Python iterator object and passed to a Scene, which is drawn using a `draw()` method. Scenes contain Mesh, Camera, and Light objects, which are applied automatically within the `draw()` call. Scenes can share objects between each other, making them useful, lightweight containers for different experimental conditions (Figure 2.6).

```

camera = rc.Camera()
camera.position.xyz = 1, 2, 3 # Move the camera
camera.projection.fov_y = 70 # Expand the vertical field of view to 70 degrees
camera.projection.aspect = 1.2 # Stretch the view to a 5/4 aspect ratio

with camera:
    cube.draw()

```

Figure 2.6: Ratcave’s Camera class. Cameras can be repositioned as with Mesh objects and applied to draw operations as with Shader objects. Changing the Camera’s intrinsic projection properties (e.g. field of view, aspect ratio, frustum section, orthographic vs perspective projection) is done through its projection attributes.

```

cube_scene = rc.Scene(meshes=[torus, cube], camera=cam)
monkey_scene = rc.Scene(meshes=[torus, monkey], camera=cam)

with rc.default_shader:
    cube_scene.draw()

```

Figure 2.7: Ratcave’s Scene class, which collect Meshes, Cameras, and Lights together and can draw them in a single Scene.draw() call.

Complex relationships of object positions can be specified via Ratcave’s simplified scene graph functionality by parenting objects to each other, allowing the experimenter to move sets of objects in a single call to the top-most parent. For example, a much-simplified solar system model could be arranged as follows (Figure 2.8).

2.4.5 Integrating Multiple Rotation Coordinate Systems

Unlike 2D objects, there are multiple ways to format rotations in three-dimensional space, three of which are: euler rotations, rotation matrices, and quaternions. Euler rotations, sequential 2D rotations about three axes stored as XYZ coordinates, have the advantage of being intuitive to use and to set; a rotation about the X axis can be written as an angle in the X rotation coordinate. However, they also come with disadvantages; for example, they must be applied in the same order every time to

```

sun, venus, earth, moon = obj_reader.get_meshes(['Sun', 'Venus', 'Earth', 'Moon'])
sun.add_children([venus, earth])
moon.add_parent(earth)
sun.rotation.y += 5 # Rotate everything about the sun’s axis.

with rc.default_shader:
    for mesh in sun:
        mesh.draw()

```

Figure 2.8: Defining Spatial Relationships via a Scene Graph. Meshes’ relative positions to each other can be set by parenting them to each other. This is done by setting Mesh.parent to the parent object.

achieve the same ending rotation (so mixing 3D modeling programs and 3D rendering programs in different order can result in unequivalent rotations), and they are vulnerable to a phenomenon known “gimbal lock”, a situation in which certain rotations can make a given axis useless. Rotation matrices, 3x3 square matrices that describe euler rotations that apply rotation transformations with a single dot product, always apply the rotations in the same order as bound by the rules of linear algebra. However, building rotation matrices by hand is cumbersome at best. Quaternions, a single rotation about an arbitrary 3-element vector, stored as WXYZ or XYZW coordinates, have the advantage of being compact, non-sequential and invulnerable to gimbal lock; however, they can be unintuitive in practice. OpenGL also requires a fourth variant, a model matrix, that has a 4x4 matrix format, whose sequential application is order sensitive. Finally, different users may prefer setting their rotations as degrees or radians!

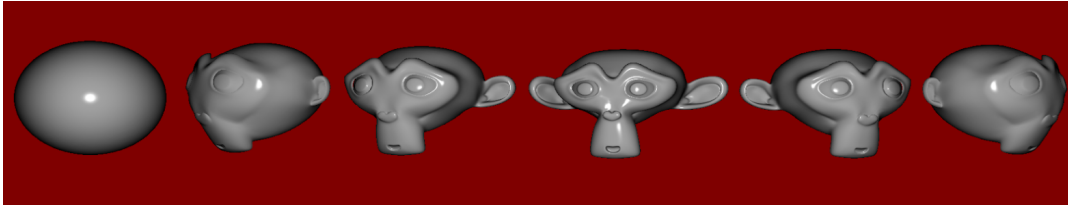
To allow full flexibility between different rotation coordinate systems, Ratcave allows all Physical object (Meshes, Cameras, and Lights) rotations to be set using any rotation coordinate system, as well as providing conversion methods between them: for example, `to_quaternion`, `to_euler`, and `to_matrix`, with options in each for setting rotation sequence and radian or degree units. This feature is, naturally, optional; by default, all rotations are specified as Euler coordinates as degrees.

2.4.6 Manipulating Data on the Graphics Card

Ratcave uses modern OpenGL constructs from the ground up, rendering by passing data to graphics card-compiled “shader” programs, rather than sending individual commands to the OpenGL state machine from Python itself. While this creates a two-language situation (Python programs on the CPU and shader programs in the GLSL language on the graphics card) that may initially seem complex, it represents a scalable solution that allows scientists to take advantage of each language’s strengths. In addition, this approach helps with creating performant 3D graphics applications in slower dynamic languages like Python, where high numbers of C library calls (common in legacy OpenGL 3D applications) can create a significant performance bottleneck; in fact, this package is used by our lab’s virtual reality system to render full 3D scenes through a multi-pass rendering pipeline at 360 fps. Three different types of data are passed to the graphics card, with each one wrapped by Ratcave with a Pythonic interface: Vertex Arrays, Uniforms, and Textures.

Vertex Arrays

Meshes in 3D applications are composed of arrays of vertex coordinates, with each defining the endpoint of an edge or the boundary of a face on that Mesh. This data could be passed to the graphics card from Python point-by-point upon drawing (OpenGL’s ‘Immediate Mode’, used by many Python 2D graphics packages), but this process can be made more efficient by sending the data as a single array using



```

mesh = obj_reader.get_mesh("Monkey", dynamic=True)
verts = mesh.vertices # an Nx3 array.

with rc.default_shader:
    while True:
        mesh.vertices = sphere_warp(verts)
        mesh.draw()

```

Figure 2.9: Warping a Mesh in real-time . If a Mesh's vertex array data is modified by the user, Ratcave will automatically send the changes to the graphics card. In this example sequence, a mesh is interpolated between its original coordinates (a Monkey primitive) and a sphere. For a video example, please see Supplementary Video 2.

OpenGL's VAO (Vertex Array Object) functionality and storing it on the graphics card itself. Sets of arrays (most commonly, a Mesh's vertex, normal, and texture coordinate arrays) can be associated together via OpenGL's VBO (Vertex Buffer Object), and then all that is needed is a single draw call when the actual rendering is performed. Since the data is already present on the graphics card, the operation is much more efficient. Ratcave pipes vertex array data using VAOs and VBOs on Meshes and uses pointers to associate NumPy arrays to the graphics card array data. The result is that users can pass NumPy arrays to Meshes and even edit them like normal NumPy arrays, while Ratcave updates the data on the graphics card as needed (Figure 2.9). Using this approach, over 30,000 vertices can be streamed in real-time to the graphics card and rendered onscreen at 60 Hz, a performance level surpassing the needs of most behavioral research studies (Supplementary Video 2).

Uniform Data

Any data that can be associated with drawing a Mesh, whether its position, color, or even custom properties that are stimulus-specific can be received by the graphics shader as so-called "Uniform" data, meaning that it has the same value across all vertices of the Mesh. Uniform data can be single values, arrays, or even matrices. Most Ratcave objects have a dictionary-like uniforms attribute that automatically links, transforms, and sends its set of uniform data whenever the object is bound (whether by calling its bind() method or using Python's with keyword) or drawn using a draw() method. Ratcave builds and maintains many uniforms automatically, including the matrices associated with positioning, viewing, and projecting objects on-screen (the Model matrix, which describes an object's position; the View matrix,

```

torus.uniforms['diffuse'] = 1, 0, 0
monkey.uniforms['diffuse'] = 0, 0, 1

scene = rc.Scene(meshes=[torus,
                        monkey])

with rc.default_shader:
    scene.draw()

```

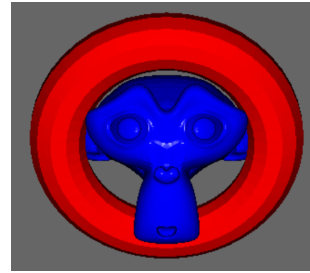


Figure 2.10: Sending uniform data to the OpenGL Shader program. Uniforms variables can be set via Ratcave objects’ uniforms dictionaries.

# Legacy OpenGL:	# Modern OpenGL:	# Ratcave
<code>glPushMatrix()</code>	<code>model_matrix = [[1, 1, 0, 7],</code>	<code>cube.position.xyz = 7, 8, 2</code>
<code>glLoadIdentity()</code>	<code>[-1, 1, 0, 8],</code>	<code>cube.rotation.x = 90</code>
<code>glScale(1, 1, 1)</code>	<code>[0, 0, 1, -2],</code>	<code>cube.scale.xyz = 1</code>
<code>glRotate(90, 0, 0)</code>	<code>[0, 0, 0, 1]]</code>	
<code>glTranslate(7, 8, -2)</code>		<code># Done during cube.draw()</code>
<code>glPopMatrix()</code>	<code>glUniformMatrix4fv(model_matrix)</code>	<code>cube.uniforms.send()</code>

Figure 2.11: Comparison of Model Matrix computation and sending to OpenGL between legacy OpenGL, modern shader-based OpenGL, and Ratcave’s interface to modern OpenGL.

which describes the camera’s position; and the Projection matrices, which describes the camera’s lens characteristics.) and adds some extra uniforms for coloring and lighting a mesh (Figure 2.10).

Besides enabling full customizability of all stimuli, using uniforms helps increase performance of OpenGL rendering pipelines in Python. Legacy OpenGL typically requires five library calls to position an object on-screen, even if the mesh’s position is unchanged from the last rendered frame. As the number of objects to be rendered increases, the computational cost scales linearly. Utilizing shaders, on the other hand, requires only a single library call, which sends a single matrix (a “Model” matrix) to the shader. Ratcave makes calculating these matrices straightforward by automatically updating the model matrix whenever a stimulus’ position, rotation, or scale attributes are modified. It also saves these transformations intelligently, “lazily” updating the matrix (via an Observer software design pattern) only when needed, using the optimized Numpy array package. Sending the matrix to the shader is done when the “draw()” method is called. Similar steps are done for the Camera’s view matrix and projection matrix (Figure 2.11).

Texture Data

Ratcave also supports mapping image data to 3D meshes using a technique called “UV mapping”, named after the coordinate system used for specifying the rows and

```

tex = rc.Texture.from_image('oak.png')
monkey.textures.append(tex)

with rc.default_shader:
    monkey.draw()
-----
tex = rc.Texture.from_image('oak.png')

with rc.default_shader, tex:
    monkey.draw()

```



Figure 2.12: Applying images to Meshes via Textures. Images, once saved as OpenGL textures, can be added Meshes in order to automatically activate them when the mesh is drawn (top-left) or activated explicitly with the *with* statement before drawing (bottom-left). Each approach has advantages for different experimental paradigms.

columns of an image (“u” and “v”). With this method, each Mesh’s vertex is associated with an image and its 2D coordinate (called a “texture” and a “texture coordinate”, respectively), and the image’s pixels are interpolated between the vertices, effectively stretching the 2D image across the Mesh’s surface.

The following three steps are needed to display this data using OpenGL: the texture data must be formatted and passed to the graphics card as an OpenGL Texture, it must be bound, and it must have an associated uniform name in order to link a given texture with a given rendering step in the shader. All of these steps are performed by Ratcave’s Texture objects, taking the OpenGL ID from an image loaded using another image processing package (making it compatible with a wide variety of image processing software), or loading it from an image file using Pyglet’s image module. If it is appended to a Mesh’s textures list attribute, it is automatically bound and its uniforms sent upon the Mesh’s draw() method call. Any number of textures of any OpenGL type (e.g. color vs depth textures, 2D vs 3D Textures, 2D vs Spherical vs 3D texture coordinates) can be appended to a Mesh, allowing any image algorithm to be implemented on the graphics card online, during stimulus rendering (Figure 2.12, Supplementary Video 3).

2.4.7 Writeable Textures: Building Deferred Rendering Pipelines.

OpenGL’s Framebuffer objects allow users to create virtual windows that redirect a rendered image to a texture saved in memory on the graphics card instead of the display. This creates opportunities to build “deferred” rendering pipelines, in which several different image processing algorithms are run and saved for a final step that combines the previous images into more complex and dynamic images (Figure 2.13).

Deferred rendering is an important technique for CAVE-style virtual reality systems, which project a 360-degree rendering of the virtual environment onto projection

```

stars = obj_reader.get_mesh('Grid3D')
stars.drawmode = rc.POINTS

monkey = obj_reader.get_mesh('Monkey')

fbo = rc.FBO(texture=rc.Texture())
with rc.default_shader:
    with fbo:
        stars.rotation.z += 10.
        stars.draw()
    with fbo.texture:
        monkey.draw()

```

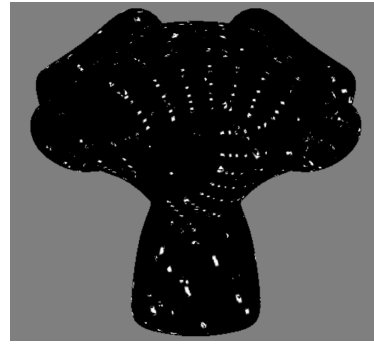


Figure 2.13: An Example of Two-Pass Rendering using Framebuffer objects. First, the stars Mesh is drawn and the resulting image saved to a Framebuffer-attached texture. This texture then becomes the texture bound to the monkey Mesh, producing an unusual effect: a rotating star field mapped on a monkey head.

```

layout(location = 0) in vec4 vertex;
uniform mat4 projection_matrix, view_matrix, model_matrix

void main(){
    gl_Position = projection_matrix * view_matrix * model_matrix * vertex;
}

```

Figure 2.14: A GLSL Vertex Shader. This program takes the vertex data (the first array in a Ratcave mesh, specified as location 0) and the various matrices (received as uniforms), and calculates the onscreen position by calculating their dot product, outputting the onscreen position.

screens surrounding the subject (Del Grosso et al, 2017). It is also useful for rendering shadows, which often entails rendering a scene from the perspective of a light source and using that image data to calculate where the shadow should appear from the camera’s perspective for the final image (see Figure 2.3 for an example).

2.4.8 OpenGL Shader Programs

Besides enabling full customization of graphics rendering, OpenGL’s programmable pipeline speeds up graphics applications by allowing users to off-load calculations to the graphics card through “shader” programs written in a C-like language called GLSL. Each program is made up of two smaller programs: one that is run for each mesh vertex (the “Vertex Shader”, Figure 2.14), which is most commonly used for positioning something onscreen, and one that is run for each pixel of the display (the “Fragment Shader”, Figure 2.15, Supplementary Video 3), which is most commonly used for setting the color and lighting properties of the image.

Ratcave Shader objects compile these programs when needed and run them when bound, as seen in Figure 2.16. Because shader programs can be mixed and matched,

```

in vec2 pos;
out vec4 final_color;
uniform vec2 mean,
uniform float width, std, theta;

void main() {
    float level = sin(pos.x / width +
        theta) / 2.;
    level *= exp(-.5*pow(pos.x-mean.x,
        2)/pow(std, 2));
    level *= exp(-.5*pow(pos.y-mean.y,
        2)/pow(std, 2));
    final_color = vec3(level + .5);
}

```



Figure 2.15: A GLSL Fragment Shader. This program is run for every pixel where a mesh is present. It takes uniform data ("width", "mean", etc) and outputs the RGB ('final_color'). This program calculates a gabor patch based on the screen position of a pixel ('pos') and the uniform parameters given by the Ratcave program.

```

shader = rc.Shader.from_files(vert='position_objects.vert', frag='gabor.frag')
gabor = obj_reader.get_mesh('Plane', position=(0, 0, -1))
gabor.uniforms['theta'] = 0.5
gabor.uniforms['mean'] = 0, 0

with shader:
    gabor.draw()

```

Figure 2.16: Loading custom shader files into a Ratcave Shader object and using it to draw a stimulus. Uniform values get sent to the shader when the draw() method is called, thereby connecting shader program variables to Python variables.

and because they run on all hardware, platforms, and graphics engines, these short programs are useful formats for a wide variety of visual stimuli.

2.5 Discussion

Whether one chooses a presentation software package, a library, a framework, or a commercial game engine, all of these tools will make use OpenGL and shader programs for their high-performance rendering. Ratcave's approach to making modern OpenGL more accessible for Python users helps bridge the gap to this powerful ecosystem, providing a cross-platform 3D graphics engine for scientists who wish to use advanced rendering methods. Since this code is similar in form to both the Python code used for manipulating arrays for data analysis, a common task for experimenters, and matches the graphics code used by graphics programmers, Ratcave users get the benefit of having a large support community in both domains. In addition, new stimuli generated for vision scientists can be published and shared in their GLSL form, allowing cross-compatibility between Python programmers using Ratcave, Matlab programmers using Psychtoolbox (which also provides Shader program-loading functions, although at time of writing no 3D scene abstraction), and 3D graphics engine users (e.g. Unity3D, Unreal Engine 4, and Panda3D).

2.5.1 Advantages of Targeted Software in the Open Source Community

Often, vision scientists searching for a stimulus presentation tool are directed to make a choice between stimulus presentation software (e.g. Neurobs Presentation, BCI2000, SuperLab, E-Prime) or stimulus presentation libraries. In practice, the distinction between these two categories can be fuzzy; for example, Psychopy can be downloaded as its own standalone Python installation and used solely through a GUI interface, and Neurobs Presentation software allows users to control the software from Python on a separate thread. In practice, this means that each of these options exists as a kind of self-contained software environment, making it challenging for one program's features to supplement another's.

All of these stimulus presentation libraries share something else with their commercial software counterparts: they are large (Figure 2.17). Rather than specialize as a specific tool, each library attempts to offer all features for all situations, even when these are technically unrelated (e.g. commercial hardware support, audio playing, experiment design, and logging). While the versatility of these large libraries can make them appealing, it also means they are highly complex, making it difficult for outsiders to contribute without first learning most of the code base first, whether to add features, enhance platform compatibility, or to fix bugs. While these libraries have been admirably maintained by their creators, it puts a large emphasis on a single maintainer to perform routine maintenance activities, rather than the innovations that originally inspired the project. Small libraries, on the other hand, reduce the cost of a fragile network by providing a smaller codebase to maintain, both making it easier for the author to provide support and for others to either fork the project themselves or even "reinvent the wheel" in case new innovations are necessary.

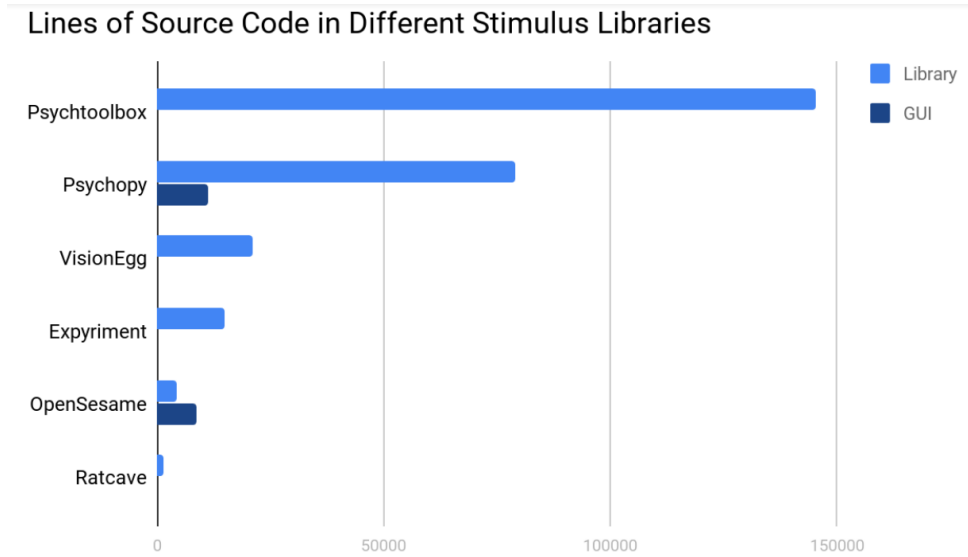


Figure 2.17: Project Size of Different Popular Stimulus Libraries. Lines of source code were counted using the cross-platform cloc program (<https://github.com/AlDanial/cloc>), with documentation and test code excluded. Code for the core library and the graphical interface are separated for libraries that also work as standalone software.

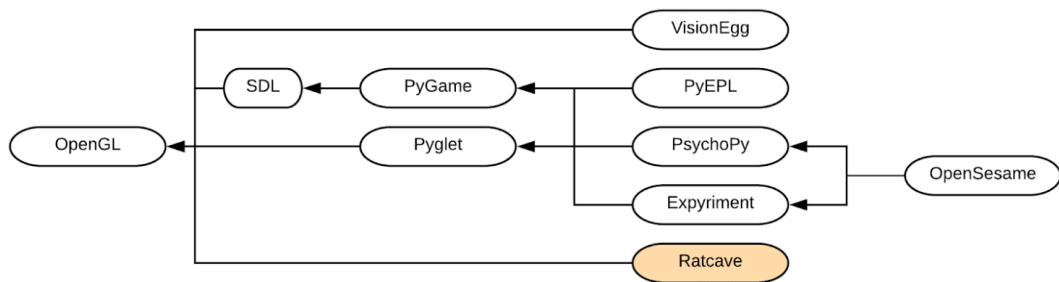


Figure 2.18: Class dependency tree of Python stimulus presentation packages.

Recently, another targeted library has been added to the psychology toolbox: OpenSesame, a Python-powered GUI experiment-building interface (Schreij & Theeuwes, 2012). OpenSesame’s use of PsychoPy, Expyriment, and PyGame as “backends” allows it to focus only on one task, and accordingly add features to a wider range of users. Ratcave is following a similar targeted approach by loosely-coupling a small but powerful set of features (3D graphics scene management and OpenGL shader manipulation) to the larger software ecosystem (Figure 2.18).

Commercial Game Engines in Place of Scientific Stimulus Packages

A third software option becoming popular in the psychology literature is the use of a commercial game engine for stimulus presentation (Jangraw et al., 2014; Solway, Miller, and Kahana, 2013). These libraries have the benefits of both of the aforementioned options; they are fast (modern 3D video games are often more demanding than typical vision stimuli), well-maintained (due to funding from the video game

industry), flexible (users write scripts with a full programming language at their disposal), user-friendly (often having full graphical interfaces for building and manipulating scenes and are highly-documented), and have a large commercial and hobbyist user base to build a community around. Not all game engines are equal, however; for example, some of these game engines are available for Python as frameworks (e.g. Panda3D, PyOgre, and Vizard), but have libraries so extensive that they both require a strong time commitment to learn to be useful and (for implementation reasons) exclude incorporation of psychophysics stimuli from other packages. For scientists who are already using one of the Python psychology stimulus packages and just want to add in 3D stimuli or customized shaders to their experiment, or for scientists who want a quick introduction to 3D graphics, Ratscave makes a good alternative to existing graphics frameworks and addition the the existing Python psychology software ecosystem.

Chapter 3

A Virtual Reality System for Freely-Moving Rodents

3.1 Author Contributions

Authors: Nicholas A. Del Grosso, Justin J. Graboski, Weiwei Chen, Eduardo Blanco-Hernandez, and Anton Sirota

N.A.D.G. designed and implemented the ratCAVE system, designed and performed behavioral experiments in the VR; J.J.G. and E.B.H. performed electrophysiological experiments recordings in the VR and behavioral experiments with real objects exploration; all authors contributed to data analysis; N.A.D.G., E.B.H and A.S. wrote the paper. All authors discussed the results and commented on the manuscript.

3.2 Introduction

Movement is a fundamental element in the action-perception loop that is critical for most cognitive functions, such as decision-making, memory and spatial navigation. Internally-driven locomotor, head and sensor movements, an exploratory repertoire of a naturally-behaving animal, allow it to actively sample sensory information from the outside world for its optimal detection and encoding, as well as guidance of the behavior (Gibson, 1966; Kleinfeld et al., 2014; Morillon et al., 2015; Gegenfurtner, 2016; Angelaki and Cullen, 2008). Recognition that the closed-loop link between internal dynamics, motor output and sensory processing gives rise to predictive coding, attention and flexible motor control (Engel et al., 2013; Adams, Shipp, and Friston, 2013; Feldman, 2015) is encouraging the use of a new experimental paradigm in sensory and cognitive neuroscience: closed-loop sensory stimulation. Traditional open-loop experimental paradigms involving head-fixation of the animal, useful for performing sensitive measurements of functional brain activity, are being replaced by experimental setups that partially close the loop between action and sensation while still retaining precise control of sensory inputs (Schwarz et al., 2010; Dombeck and Reiser, 2012; Minderer et al., 2016; Guo et al., 2014; Carandini and Churchland, 2013).

Virtual reality (VR) systems close the loop between locomotion and vision. Many rodent laboratories use head- or body-restrained VR (rVR) setups to simulate locomotion through a 3D virtual environment (VE) via running on a treadmill (Dombeck and Reiser, 2012; Minderer et al., 2016). Spatial coding research has especially benefited from such systems; VR researchers have taken advantage of the flexibility of a VE by implementing arbitrarily-large environmental exploration paradigms utilizing dynamic environments (Hölscher et al., 2005; Jacobs et al., 2013; Thurley et al., 2014) and manipulating visuomotor gain (Chen et al., 2013). Additionally, many researchers take advantage of the rodent's fixed head by performing optical and intracellular recordings during locomotion through virtual space, a normally-challenging task in freely-moving animals (Dombeck and Reiser, 2012; Schmidt-Hieber and Häusser, 2013; Domnisoru, Kinkhabwala, and Tank, 2013).

However, locomotion on a treadmill alone may not be enough for performing closed-loop research; behavioral and physiological differences between rVR and real-world navigation illustrate the detrimental effect of sensorimotor loop disruption and the importance of increasing motor affordances. While head-fixed rodents in rVR experiments are limited to navigating linear tracks (Dombeck and Reiser, 2012; Harvey, Coen, and Tank, 2012; Domnisoru, Kinkhabwala, and Tank, 2013; Chen et al., 2013; Schmidt-Hieber and Häusser, 2013), likely due to an impoverished sensory-motor loop (Schmidt-Hieber, personal communication), rodents can navigate a two-dimensional VE if only their bodies are restrained and their heads left free to move (Hölscher et al., 2005; Thurley et al., 2014; Aghajan et al., 2014; Cushman et al., 2013;

Aronov and Tank, 2014). If rats are further allowed to rotate while running on a spherical treadmill in rVR experiments, 2D hippocampal place cell representation of the VE is comparable to that in real-world navigation (Aronov and Tank, 2014); however, this effect is lost if the rodent's body rotation range is limited (Ravassard et al., 2013; Aghajan et al., 2014).

Despite the utility of rVR for studies of spatial navigation, animal restraint still poses unresolved challenges. First, restrained animals exhibit constrained or limited behavioral patterns within 2D space, which affects the way they actively sample the 3D environment. Second, locomotion-driven visual input is in conflict with locomotion-independent, head-bound idiothetic, olfactory, tactile and auditory inputs. Third, proprioceptive and vestibular inputs in rVR setups are diminished and unnatural, making them potential causes of the observed reduction in frequency- and speed-correlates of theta oscillatory dynamics, compared to rodents allowed to freely navigate the real world (Ravassard et al., 2013; Aronov and Tank, 2014). Finally, animals require long and complex training and habituation to rVR setups (Aronov and Tank, 2014; Thurley and Ayaz, 2016).

These challenges are resolved if visual feedback in VR is based on head motion in 3D space in freely-moving subjects, giving rise to a coherent visual, idiothetic and external multisensory input, an unperturbed action-perception loop, and a full repertoire of rodent behavior, while still preserving the precise control of visual stimuli in VR setups (Scarfe and Glennerster, 2015). One such freely-moving VR (fmVR) system was introduced for human subjects as the Computer-Assisted Virtual Environment (CAVE) (Cruz-Neira et al., 1992). A CAVE allows observers to freely move in space and view a 3D VE on the projection surfaces surrounding them. To date, CAVE-like VR systems for flies (Strauss, Schuster, and Götz, 1997; Fry et al., 2009) and fish (Orger et al., 2008) couple animal 3D motion to 2D contrast patterns on the projected onto cylindrical surfaces, though a system for arthropods with more realistic visual feedback was reported (Stowers et al., 2014; Stowers et al., 2017). Implementation of the CAVE system in rodents, a model mammalian system where complex interrogation and manipulation of the nervous system can be combined with cognitive behavior, would open new dimensions in experimental neuroscience. The development of a next-generation, immersive fmVR was called for in a recent review (Minderer et al., 2016). Designing an immersive fmVR in quickly-moving animals is challenging, however, as it would require very-low-latency visual feedback to avoid introducing new conflicts in the sensorimotor loop (Ash et al., 2011) and computationally-intensive graphical operations to produce a visually-rich VE.

To provide an immersive virtual environment for untrained freely-moving rodents and allow them to explore and interact with the virtual environment in a natural

manner, we developed a new CAVE fmVR system (Ratcave VR) that produces minimal inter-sensory conflict during self-motion using fast head-tracking and high display frame rates, as well as enriched visual 3D cues of the virtual scene. We demonstrate the naturalistic interaction of rats with VEs in our fmVR system in several behavioral tasks. We further show a use case of fmVR not possible with rVR systems: to study the multisensory nature of hippocampal spatial representation. This highly-immersive fmVR system can be a powerful tool for a broad range of neuroscience disciplines.

3.3 Results

3.3.1 Ratcave VR: VR system for freely moving rodents

We implemented a CAVE system where a VE projection on the surface of the arena was closed-loop coupled with the real-time tracking of the head of the animal. In this setup, animals could move freely in a rectangular arena similar to that used for conventional open-field experiments, with the VE front-projected onto the arena's surface. We used an array of 8 high-speed cameras (240 fps, NaturalPoint Inc.) to track the 3D position of the rodent's head via a rigid array of retro-reflective spheres attached to a head-mounted 3D-printed skeleton (Figure 3.1c,d). This tracking system enabled us to record the rodent's head position with very high spatial (<0.1 mm) and temporal (<4.2 msec) resolution. The VE, designed using the open-source 3D modeling software Blender3D, was rendered each frame in a full 360-degree arc about the rodent's head and mapped onto a 3D computer model of the arena using custom Python and OpenGL packages (Supplementary Figure 3.7, Online Methods) and warped in real-time to generate a fully-interactive, geometrically-accurate 3D scene (Figure 3.1b). The core cube-mapping algorithm used to perform the mapping of the VE onto the projection surface was identical to those described in rodent rVR setups (Supplementary Figure 3.8a-c) (Aronov and Tank, 2014), with the addition of a VE projection that is continuously updated according to the changing 3D position of the rodent's head (Figure 3.1b), resulting in perception of a 3D VE that is stable in the real-world frame of reference the animal moves about (Figure 3.1c-d). Because the presented virtual motion parallax cue automatically takes into account the rodent's distance from the arena's walls, virtual objects can be made to appear both inside and outside the arena's boundaries (Supplementary Movie 1).

3.3.2 Flexible design, calibration and mobility of the VR arena

Automatic arena-projector calibration ensured that the image was correctly projected onto the arena's surface. Calibration was realized via a point cloud-modeling procedure by projecting a random dot pattern onto the arena's surface, measuring the 3D position of each dot via a 3D tracking system, and fitting a 3D digital model of the arena to this point cloud data (Figure 3.1a). This scanning process provides the flexibility to layer a VE over an arbitrary arena surface, including smooth objects inside the arena. The position of the arena with respect to the projector was continuously tracked using a set of retro-reflective spheres mounted on the arena itself, allowing the arena to be arbitrary translated and rotated during an experimental session while preserving the correct projection.

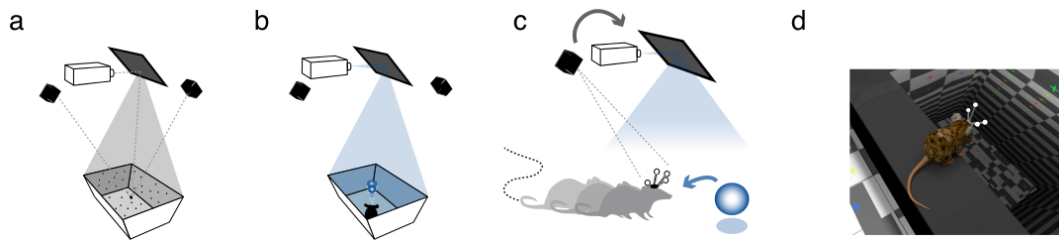


Figure 3.1: Schematic of the Ratcave VR components. (a) Projector-arena mapping scheme. A dot pattern is projected onto the arena's surface, and the 3D point cloud, detected by the motion tracking camera array, generates a matching 3D mesh of the arena and calibrates the video projector. (b) The virtual environment is front-projected via the mirror onto the arena surface. (c) The freely-moving rodent's head position is tracked by means of a head-mounted array of retro-reflective spheres using a multi-camera 3D tracking system. (d) A 3D virtual environment (e.g. a virtual cliff.) can then be rendered from the rodents perspective.

3.3.3 Low latency motor-visual feedback of the Ratcave VR system

Motion-to-photon (end-to-end) latency in our system cumulatively included input lag of the tracking system, the processing lag of the tracking and Ratcave VR software, as well as "display lag", the time it takes for the rendered image to be projected. Selecting fast tracking and display hardware and optimized software allowed us to achieve a motion-to-photon latency approaching 15 msec (Figure 3.9a-c). This latency is significantly lower than that of any fmVR/CAVE systems reported to date that we are aware of and additionally supplies a smoother motion stimulus than in those with lower-framerate displays (typically 60 Hz) (Kim et al., 2015; Stowers et al., 2014). Since rats rarely reached speeds of 50 cm/s during spontaneous exploration of the arena (Figure 3.9d), we expect that they were experiencing minimal, if any, latency-related cross-sensory conflicts in our system.

3.3.4 Visual cues enhancing VR immersion

A large number of conflicting visual cues can exist in CAVE systems that can distract from VR immersion, which we've taken additional steps to decrease. First, we implemented online radiosity compensation, which equalizes the image brightness across the entire arena to decrease the visual perception of the arena itself. Second, we implemented antialiasing to decrease the perception of the individual pixels. Third, the location of the virtual light source was programmed to match the position of the projector, giving the projector the impression of simply illuminating the virtual objects, rather than creating them. Finally, to provide a richer visual scene and additional visual depth cues to the observer (Hu et al., 2000), we implemented both diffuse and "glossy" specular reflections off the virtual objects' surfaces using the Phong reflection model, as well as casting shadows on themselves and other objects. Additions of these visual features gave rise to a smooth and perceptually realistic VE (Supplementary Figure 2d).

3.3.5 Testing spontaneous behavior of rats in the Ratcave VR

We designed a set of behavioral experiments that were aimed to explore and evaluate the degree of rats' immersion and interaction with the VE provided by Ratcave VR. In each experiment, the behavior of freely-moving rats ($n=9$) was tested in distinct VEs that were designed to evaluate specific aspects of behavioral interaction with purely virtual elements: virtual cliff avoidance, virtual object exploration, and interaction with a virtual wall. These tasks were specifically chosen to require no pre-training or reinforcement and rely on spontaneous behavior of rodents. Benefiting from high spatial resolution tracking of position and orientation of the rats' head, each rat's natural behavior during each task was classified into walking, immobility and rearing based on speed and head-height features (Supplementary Figure 6a). The three experiments were performed repeatedly across animals over several days.

3.3.6 Virtual cliff avoidance experiment

The visual cliff avoidance paradigm is a classical test of visual depth perception and relies on the subject's innate behavior (Walk and Gibson, 1961). We designed a virtual version of this task that tests if rats avoid jumping from the virtual cliff emulated in the VE. In each 30-second session, rats were placed onto a random end of a board suspended above the arena's floor, bisecting the arena into a randomly-assigned safe side and cliff side, in which the virtual floor was either at floor level or 1.5 meters below the floor level, respectively (Figure 3.2a; Supplementary Movie 3, detailed experimental methods can be found in Supplemental Materials). Three different types of cliff stimuli were used in this experiment: a "Real" cliff (following the class visual cliff paradigm, with a transparent, nonreflective acrylic floor over a checkerboard-lined well), a "Virtual Cliff" (the same well, simulated by projection onto the arena floor using the VR system), and a "Static" cliff (a cliff VR projection that did not update as the rat changed position), Figure 3.2b. We observed several well-defined behaviors in this task: wall-supported rearing, visual exploration of the ledges (head dipping), and the jump off the ledge towards one of the virtual floors (Figure 3.2c). We found that rats showed a preference toward the safe side of the arena for both the Real and Virtual cliff conditions, but not for the Static cliff (Figure 3.2d). This result was consistent with the visual cliff literature.

3.3.7 Interaction with virtual walls

Virtual boundaries are the main elements of the VE that inform animals about the topology of the virtual space (Barry et al., 2006). In rVR systems, rats are traditionally operantly conditioned to respect the boundaries by freezing the VE upon collision of the animal's virtual trajectory with the wall (Aronov and Tank, 2014; Thurley et al., 2014; Aghajani et al., 2014). In order to investigate how naive rats spontaneously interact with virtual boundaries, we introduced a virtual wall that bisected the arena in one of four randomly-selected locations (Figure 3.3a-b). Rat exploration

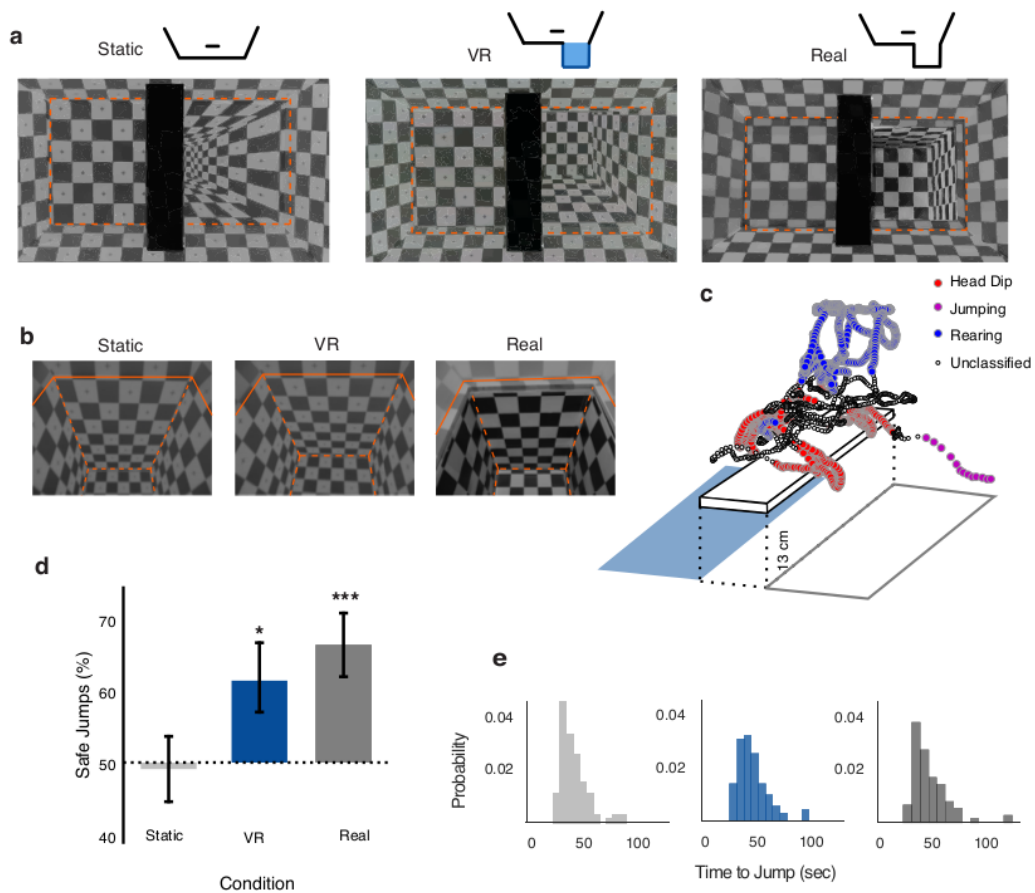


Figure 3.2: Virtual Cliff Avoidance Experiment. (a) Schematic section and image of the arena for each experimental condition: Static Cliff Image, Virtual Cliff, and Real Cliff. (b) Close-up pictures of the three conditions from the rat's perspective, orange lines highlighting the cliff edges. (c) Example trajectory and segmentation of the rat behavior during a single session. (d) Ratio of safe-side jump trials by condition. Error bars represent 68% confidence intervals. (e) Frequency of the time prior to jumping for each condition. Asterisks represent p values below different alpha thresholds (*: $p < 0.05$; ***: $p < 0.001$).

patterns (occupancy times, speed changes, locomotion direction, rearing locations, and trajectory changes) were compared between phases in which no virtual wall was present (pre- and post- stimulus phases, 4 minutes each) and a phase in which the wall was present (also 4 minutes in duration). Rats displayed a noticeable change of their behavior in the vicinity of the walls, as demonstrated by increased occupancy and rearing events around the wall (Figure 3.3c,d, Supplementary Movie 3). Orientations of the locomotion trajectories in the vicinity of the virtual wall concentrated around perpendicular and parallel orientations to the wall (Figure 3.3e,f), indicating that the rat moved either along or towards/away to the virtual wall. This behavior is consistent with thigmotaxis along both virtual and real walls. We further tested whether rats treated the virtual wall as an obstacle when approaching it. Locomotion trajectories approaching the virtual wall were more likely to change direction than to maintain direction when the virtual wall was present, compared to the same arena locations in the pre- and post-stimulus phases (Figure 3.3g). Thus, rats' interactive behavior towards the virtual wall is consistent with them responding to it as a wall.

3.3.8 Exploration of virtual objects

Spontaneous exploration of objects is the cornerstone for multitude of behavioral paradigms aimed to study perception and memory (Blaser and Heyser, 2015). Real objects have multimodal features and affordances, but require careful and laborious handling for repeated presentation and feature manipulation. 3D virtual objects could be arbitrarily designed, manipulated and presented to an animal automatically. While rodents can perceive 3D shapes (Zoccolan, 2015) and navigate towards reward locations marked by virtual objects in rVR (Aronov and Tank, 2014; Cushman et al., 2013), naturalistic exploration of virtual objects cannot be properly tested with any existing methods. In series of test sessions, we investigated how rats spontaneously interact with virtual 3D objects pseudo-randomly positioned on one end of the arena, as compared to a control location on the opposing end (Figure 3.4a-c; Supplementary Movie 3). To control for sensorimotor conflict and isolate vision-mediated exploration, all objects were projected so that they were located under glass domes, unreachable to the rats. To ensure the generality of our findings, we also used real, 3D printed plastic objects in pseudorandomly-selected sessions. Finally, we controlled for rat's exploration biases by comparing occupation preference between within-session phases (pre-stimulus, stimulus, and post-stimulus, 4-min each, as in the virtual wall experiment). We found that rats spent more time in the vicinity (<12 cm) of the virtual objects (Figure 3.4d).

3.3.9 Effect of virtual environment on hippocampal spatial map

While, as we've shown above, animals immersed into the VE interact with it less reliably than with a real environment and thus the behavioral readout only partially

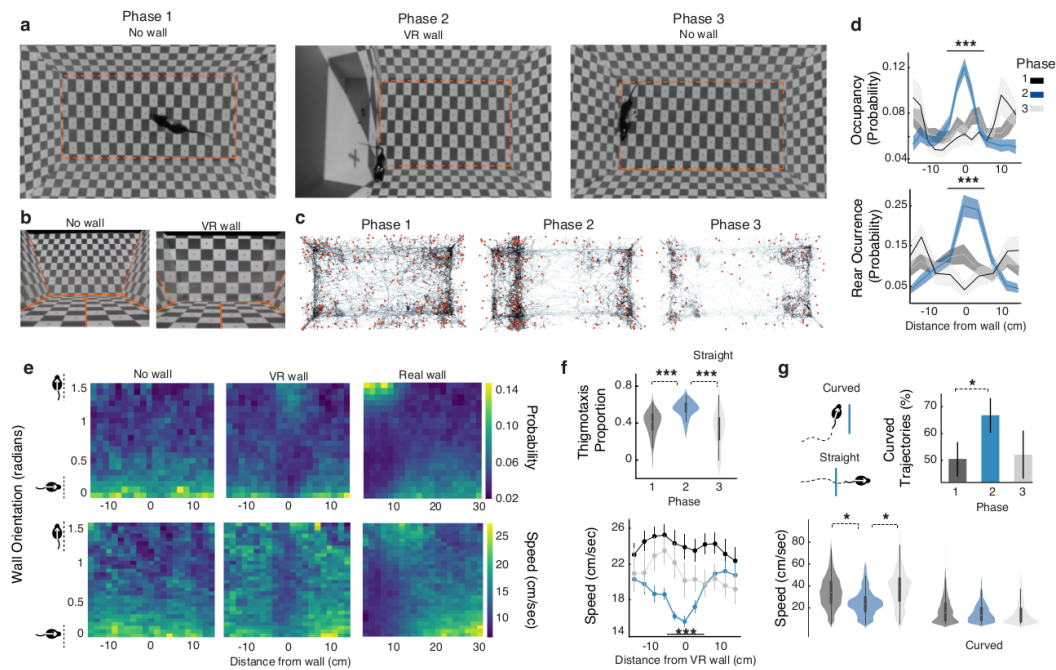


Figure 3.3: Virtual Wall Exploration Experiment. (a) Experimental protocol. The 4-minute Pre-Stimulus, Stimulus, and Post-Stimulus phases of each session. (b) Rat perspective images of the arena and virtual wall during the Pre-Stimulus and Stimulus phases. (c) Example rat trajectories from a single session for each phase; red dots mark the location of rearing events. Virtual wall position matches the protocol image. (d) Mean Session distribution of near-wall occupancy time and rearing events for each phase, both showing an increase during the Stimulus phase. (e) Upper: rat locomotion direction with distance from the virtual wall (Pre-Stimulus and Stimulus phases) and the arena walls (All Phases). Rats were more likely to move parallel to both virtual and real walls. Lower: Same as upper, with color representing mean locomotion speed. (f) Summary statistics for locomotion direction (upper) and speed (lower) by condition and distance from virtual wall. (g) Upper, Speed distributions for curved and straight trajectories. Wall-crossing trajectories were most likely to curve and straight trajectories were slower during the Stimulus phase. Error bars represent 68% confidence intervals. In violin plots, shaded regions represent the data's distribution. Asterisks represent p values below different alpha thresholds (*: $p < 0.05$; ***: $p < 0.001$).

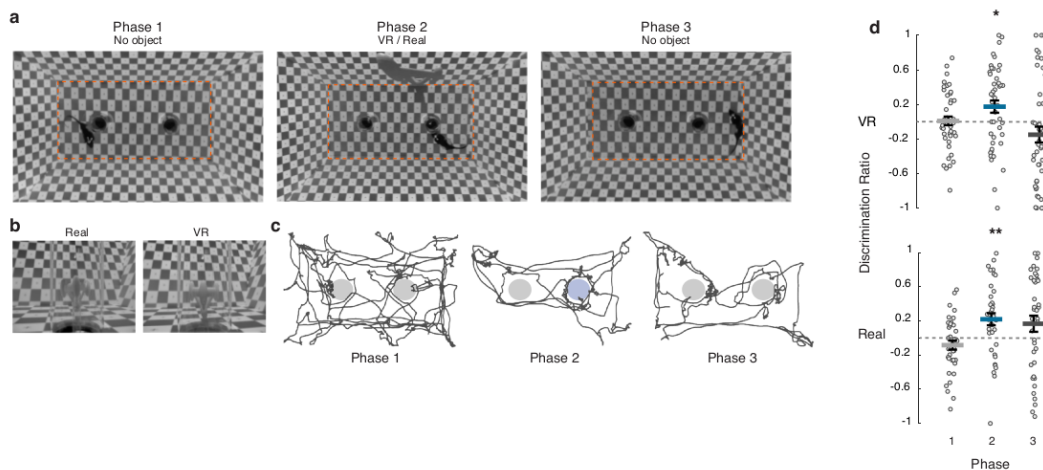


Figure 3.4: Virtual Object Exploration Experiment. (a) Experimental protocol. Each session consisted of three 4-minute phases: a Pre-Stimulus, Stimulus (object in left or right transparent dome), and Post-Stimulus phase. (b) Images of real and virtual object from rat's perspective. (c) Example rat trajectories from a single session for each phase. (d) Discrimination index of exploration preference for object vs control dome (Positive: object dome; Negative: control dome) for each session, by phase and object type (median \pm SEM). Object exploration increased during the Stimulus phase, as compared to control position. Asterisks represent p values below different alpha thresholds (*: $p < 0.05$; ***: $p < 0.001$).

reflects the animal's perception of the VE, their internal hippocampal representation of the virtual space could provide an insight into the animal's perception of the VE (Aronov and Tank, 2014). Hippocampal spatial representations are believed to be anchored to multiple frames of reference, which are concurrently controlled by visual geometrical features of the boundaries and landmarks, other external sensory and idiothetic inputs, but due to physical limitation of the real environment, dissociation of the contribution of these different reference frames is difficult, and was so far mainly limited to rotations around a symmetry axis (Knierim and Hamilton, 2011). Here, we illustrate an application of the Ratcave VR to study complete dissociation of visual and all other multisensory systems on hippocampal spatial representation by linearly translating visual boundaries with respect to the physical environment. In this pilot experiment, we recorded population of pyramidal cells in CA1/2 regions of the hippocampus (166 and 154 from two days analyzed) in a rat spontaneously exploring the arena through series of sessions in which VE whose walls were either aligned with the arena ("Normal" condition) or laterally shifted by 20 cm with respect to the physical boundaries of the arena ("Shifted" condition), Figure 3.5a. Twenty pyramidal cells were identified as simple place cells and kept for analysis based on their unimodal peak firing rate. Similar to the virtual wall interaction experiment, the rat interacted with the virtual boundary that appeared inside the arena in the Shifted condition at least during the first Shift session. Interestingly, the population of place cells ($n=20$, see Online Methods for selection criteria) remapped

their place fields within the arena between Normal and Shifted sessions in the direction of the VE shift (Figure 5b-c, Kruskal-Wallis $H=35.40$, $p < .001$ and $H=5.92$, $p=.21$, length and width, correspondingly). Between-session shift differences were probed using post-hoc Wilcoxon paired-rank tests (Normal1-Shift1: $W=12$, $p < .01$; Shift1-Normal2: $W=15$, $p < .01$; Normal2-Shift2: $W=56$, $p=.11$, ns). No shift occurred when the "StarField" environment was presented ($W=97$, $p=.77$, ns). The effect decreased over consecutive alternating sessions and following multiple exposures to the shifted VE (3 days later) place cells showed no remapping between Shifted and Normal conditions (Figure 5d-e). We tested if any visual information associated with VE boundaries is contributing to the stabilized spatial map by immersing the rat into the VE that was unrelated to and expanded beyond the physical boundaries of the arena. This VE as well had no effect on the place field position (Figure 5d-e, bottom). Thus, Ratcave VR is sufficiently immersive that it enables visual input control of hippocampal spatial representation, but progressive exposure to the conflict between visual and other multisensory inputs enabled by Ratcave VR can result in complete independence of the hippocampal spatial representation from the visual input (Jeffery and O'Keefe, 1999; Geva-Sagiv et al., 2016).

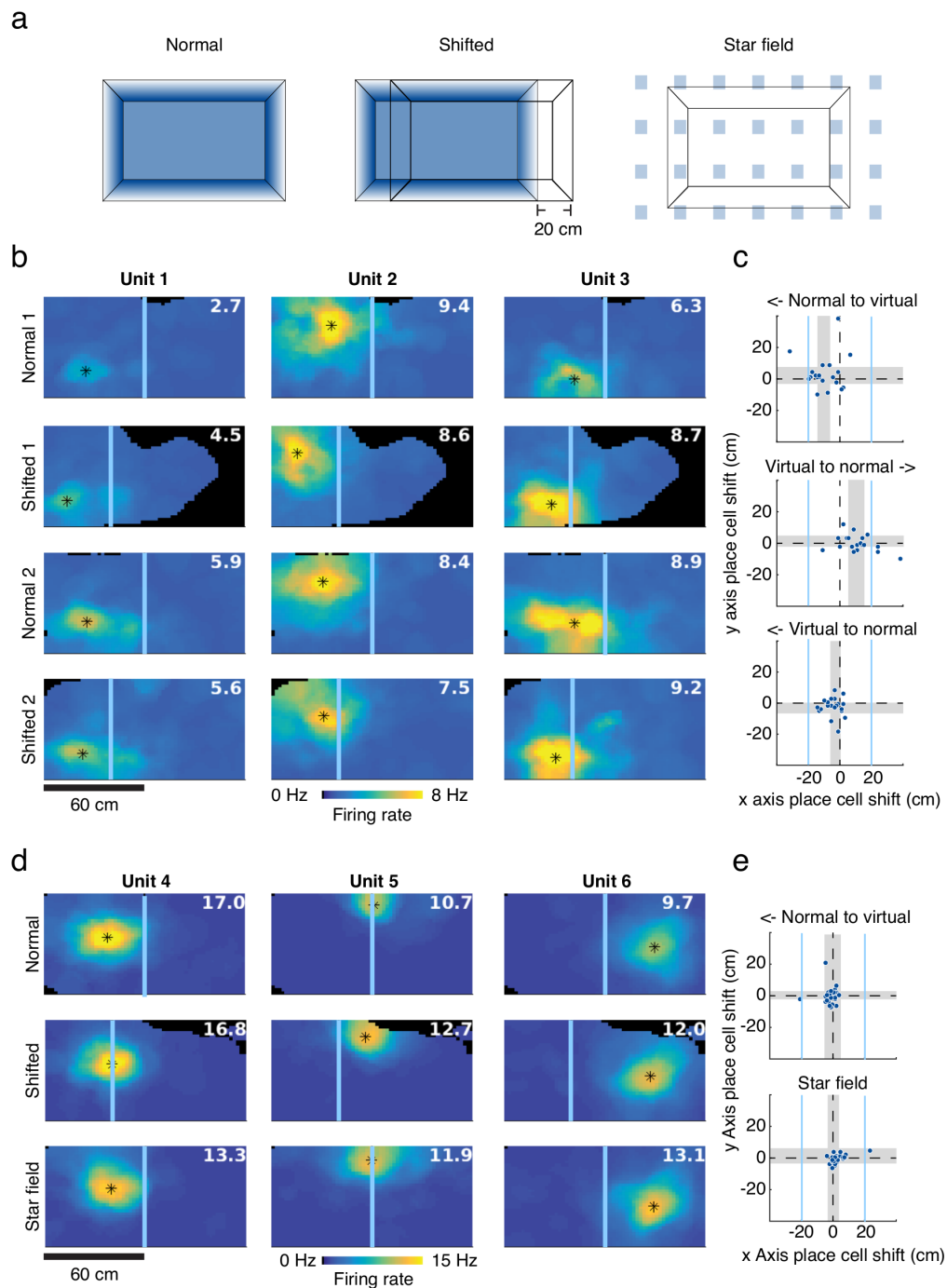


Figure 3.5: Impact of virtual environment on hippocampal spatial representation. (a) Schematics of the virtual environments used in the experiment: "Normal" (left), "Shifted" (Center), and "Star Field" (right). (b) Examples of place fields of hippocampal place cells, with center position of the field marked by an asterisk and center of virtual arena shown as a blue bar. Rows show session progression of alternating virtual environment. White numbers indicate the cell's peak firing rate (spikes/s). (c) Analysis of the place field center shift between conditions. Scatterplots indicate X- and Y-axis shift of the location of center place fields between each consecutive session pair, with 95% confidence interval of population shift estimate ($n=20$) indicated as gray shading. Non-overlap of gray bar with dotted line indicates a significant place field shift in the given axis. (d-e) Place fields for units recorded three days later, same as in b and c.

3.4 Discussion

We presented a CAVE system for freely-moving rodents that builds on and extends previous developments of fmVR systems in arthropods and fish (Strauss, Schuster, and Götz, 1997; Fry et al., 2009; Orger et al., 2008; Stowers et al., 2014). This system is a general-purpose cognitive science VR research platform by implementing a combination of methods that provide realistic visual environments, low-latency and high-precision closed loop feedback to animals' head, and flexibility of the shape and mobility of the arena. Using more complex lighting models, including diffuse and specular reflections and self-shadowing, provides new visuo-spatial cues for virtual environments and increases immersion (Hubona et al., 1999; Hu et al., 2000). In humans, sensory conflicts resulting from out-of-phase feedback to rapid head motion arise when motion-to-photon latency of the VR system is larger than ca. 50 msec, resulting in decreased performance in spatial navigation, spatial perception, and sense of self-motion in the VE (Ash et al., 2011); to counter this effect, we've implemented a low-latency visual update loop (240 fps, 15msec "motion-to-photon" lag) to decrease mismatch between vestibular, proprioceptive, and visual self-motion cues, essential for proper self-motion detection and functioning of the head-direction system (Warren and Hannon, 1988; Taube, 2007).

There are pressing improvements needed to further increase immersion in VR systems used in neuroscience research. While rVR immersion requires animals to ignore lacking or mismatching sensory inputs, immersion in fmVR is associated with the minimal conflict between visual and other senses. However, both rVR and fmVR systems suffer from the cross-sensory conflict upon collision of animal's trajectory with the virtual boundary and can break immersion. In rVR setups, the solution has been to simply stop visual update while still allowing rodent locomotion, creating a locomotion-visual mismatch upon impact (Aronov and Tank, 2014; Thurley et al., 2014). In fmVR, a similar mismatch occurs when the virtual and real surfaces are not matched and are directly sampled by the animal. Such situations require a careful selection of virtual environment, arena design and method to match the research questions at hand. A few improvements can be considered in the Ratcave VR. First, VE objects and boundaries can be made inaccessible to the animal by projecting them outside arena walls or across the gap. Second, Ratcave VR calibration procedure allows for projecting virtual objects on smooth shapes inside the arena, thus aligning them with real counterparts, enhancing VR immersion via all three avenues: naturalistic interaction (via touch and smell), increased cue salience, and reducing cross-sensory mismatch upon virtual object contact. Third, electrical or optogenetic stimulation of olfactory or somatosensory system (Smear et al., 2013; Sofroniew et al., 2015) can be used to provide congruent multisensory feedback. Similarly, use of visuo-acoustic VR can be provide more cohesive VE (Seeber, Kerber, and Hafter, 2010; Cushman et al., 2013). In addition to motion-dependent monocular depth cues, static binocular depth cues based on stereoscopy are also important for forming an

accurate 3D space percept (Hubona et al., 1999), a point currently ignored in rodent VR studies. Thanks to precise head-based projection, the Ratcave VR system can be extended since the completion of the experiments described in this work to generate stereo VE via implementation of both anaglyph and head-mounted shutter glasses to systems provide different images to the left and right eyes of the exploring rodent. Many of these improvements can be added onto existing rVR and fmVR systems to increase VR immersion in those setups. Further integration and cross-insemination of open-source fmVR and rVR developments in diverse animal models will enable a broad spectrum of neuroscientists to use these systems.

Freely-moving virtual reality represents an improvement in VE immersion over rVR, considered as an enhancement of naturalistic interaction mechanisms with the virtual environment, an increased salience of sensory cues associated with the virtual environment, and a minimization of cross-sensory conflict. Naturalistic interaction with the virtual environment is enhanced in fmVR by simply allowing the full range of movement in an unmodified space, without training or postural alteration, while in rVR, locomotion and virtual object interaction must be simulated via running on a spherical treadmill. Self-motion cues through the virtual environment are enhanced in fmVR by providing higher-frequency and shorter-latency feedback to head motions in the virtual environment alongside the lower-frequency locomotion behaviors, while rVR only provides locomotion feedback. In contrast to rVR that assumes a stationary head in the virtual projection, fmVR system minimizes cross-sensory conflict by providing feedback to head motions, as well as by matching changes in olfactory, tactile, and auditory real-world inputs to self-motion in the virtual world. Finally, fmVR systems do not require operant training and habituation procedures used in rVR systems.

We demonstrated that a Ratcave VR system for freely moving animals can be successfully applied to a number of behavioral paradigms not possible with conventional rVR systems. Untrained rats freely behaved and spontaneously interacted with virtual environment by approaching, exploring and leaving virtual objects and walls, displaying thigmotaxis along virtual walls and avoiding a virtual cliff. We further used Ratcave VR system to illustrate how contribution of the virtual visual input to hippocampal spatial representation can be strong upon first exposure to VE mismatched with the physical world, but becomes negligible after repeated exposure of the rat to cross-sensory conflict. These experiments and design features of Ratcave VR described above pave the way to a large body of future applications.

First, high spatio-temporal resolution of 3D tracking of the rodent's head, which can be extended to include the full body, enables quantitative analysis of the natural behaviors of the rodent during VE exploration, which significantly extends level of analysis possible using two-dimensional locomotion information provided by conventional tracking in 2D space or the treadmill measurement in rVR. Second,

Ratcave VR's "trackable" arena also enables vestibular perturbations during VR experiments via arena movement, enabling studies on vestibular system function and visuo-vestibular binding in behaving rodents. Third, fmVR's ability to incorporate a three-dimensional element into operant conditioning tasks increases the range of motor affordances of digitally-rendered learning stimuli, which have their own benefits of flexibility and timing control (Furtak et al., 2009). Integrating these improvements into VR setups will enable new methods in research areas such as learning and memory, perceptual decision-making, and 3D-rotation and object perception (Zoccolan, 2015). Fourth, the automated nature of head tracking allowing for online behavior analysis, operant conditioning, and fmVR enable high-throughput and automatic behavioral testing in a colony of animals (Schaefer and Claridge-Chang, 2012) across a large variety of tasks, such as perceptual, incidental and motor learning, spatial memory paradigms, to name a few. Importantly, use of automated fmVR behavioral paradigms allows their standardization, reproducibility of results independent of experimenters or setup. Finally, combined with neural recording and manipulation Ratcave VR enables the detailed investigation of the mechanisms of spatial coding. Manipulation of the arena boundaries provides a powerful tool to study for multisensory nature, remapping and attractor properties of the spatial representation (Jezek et al., 2011).

Low latency, unmatched by any other system for freely moving subjects, and rich visual features make Ratcave VR appealing for use in human subjects. Translation of experimental paradigms and physiological validation of psychophysical experiments from humans to animals and back could enable validation and further development of diagnostic and rehabilitation procedures for the vestibular or neurodegenerative disorders in animal models (Bergeron, Lortie, and Guitton, 2015; Fritz, Cheek, and Nichols-Larsen, 2015). Ratcave VR opens new ways to study sensory-motor systems in their natural dynamics while having flexibility in manipulating the sensory feedback not possible in real life.

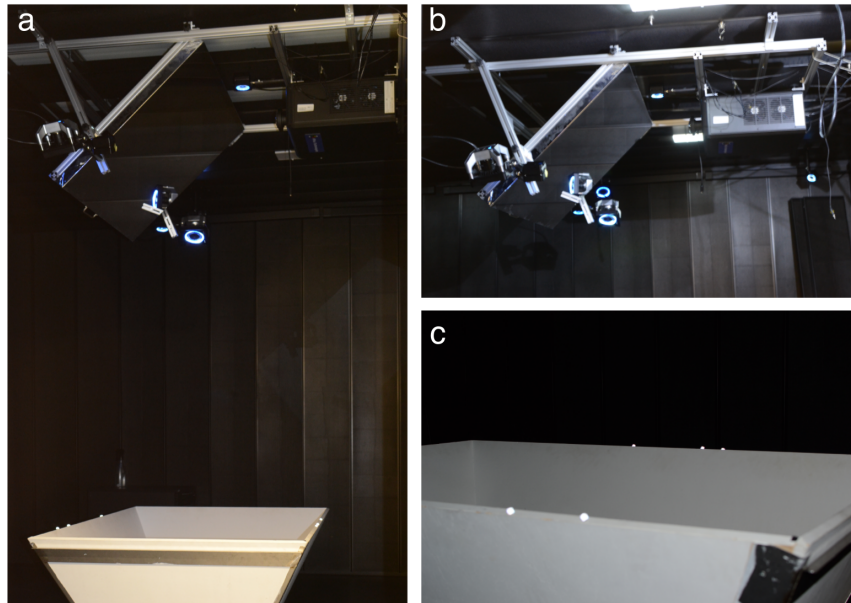


Figure 3.6: Photographs of the Ratcave VR setup. (a) The full system, including showing arena, projector, mirror, and cameras (shown with blue lighted rings, normally turned off). (b) Close-up on the projector, mirror, and cameras. (c) Close-up on the arena showing retro-reflective markers attached; the increased brightness of the markers is created in the photo by the camera's flash. Infra-red LEDs, invisible to rodents, light these markers during tracking.

3.5 Online Methods

3.5.1 Ratcave VR

Hardware Setup

Our setup consisted of a rectangular arena with dimensions 115cm x 65cm (L, W) and walls 40cm high, angled at 70 degrees to increase the projected image's surface area and brightness. A set of 9 cameras (OptiTrack, Prime 13, NaturalPoint Inc. U.S, 240 fps) was used to record the 3D position of retro-reflective spheres. A projector with a 240 fps frame rate (VPixx Technologies Inc., Saint-Bruno, Canada) was mounted to the ceiling. An optically-flat aluminum-foil projection mirror (100cm x 75 cm, Screen-Tech), slanted 45 degrees, was suspended from the ceiling on an adjustable frame for accurately fitting the projected image onto the whole surface of the arena. This setup was installed inside an isolating acoustic chamber (Supplementary Fig. 4).

Software

The Ratcave VR system depends on many pieces of software to work; interactions between each software component are diagrammed in Supplementary Figure 3. Virtual environments are modeled and exported to file in a 3D modeling program, Blender 3D (Supplementary Fig. 3a). Coregistration of the arena and projector

with the tracking coordinate system is performed via a custom Python command-line program package called "Ratcave VR_calibration", which uses a custom Python API called "MotivePy" to access and controlling our Optitrack camera array while using a custom Ratcave VR Python 3D graphics package to render the point cloud from the projector (Supplementary Fig. 3, "Grey Zone"). Ratcave VR provides a user-friendly interface for modern OpenGL rendering techniques, and its "Get Data, Update Camera, Render VE" event loop forms the core engine of a Ratcave VR virtual reality session. Cubemapping, lighting, and antialiasing are done via OpenGL FrameBuffer objects and shader scripts supplied with Ratcave VR. VR Experiment scripts are written in Python, using a custom network client called "NatNetClient" to obtain Optitrack camera data in real-time and Ratcave VR to render the virtual scene (Supplementary Fig. 3, "Blue Zone"). Because all software used in the Ratcave VR VR setup is comprised of loosely-connected specialized parts, the software developed by the lab is generalizable to a variety of different setups, enabling other labs to substitute like-components to build a VR setup that matches their hardware.

Code Availability

All code used in implementation of the Ratcave VR is freely available for use and modification via Github at the Ratcave VR organization (www.github.com/RatcaveVR) and installable via the Python Package Repository. Associated documentation and usage tutorials are available at RatcaveVR.readthedocs.io and in Supplementary Documentation. All custom-written code used for analysis of results presented in this study are available from the corresponding author on reasonable request.

Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

VR implementation

We tracked the rodents' head position and orientation by mounting a 3D-printed plastic skeleton of four retro-reflective spheres (6-8mm in diameter). Commercial software (Motive, NaturalPoint Inc., USA) isolated these spheres' positions in each camera's imaging data and reconstructed the 3D position and orientation of the rigid body (Figure 3.7, "3D Tracking Software"). Rodent head position was then logged for offline analysis and sent over the network to the VR system's experiment script via a custom python package (NatNetClient) for visual stimulus update (Figure 3.7, "Python Optitrack Client"). The Ratcave VR VR engine Ratcave VR receives the current position of the rat's head from NatNetClient, updating the virtual scene from the rat's perspective, generates the projected image using a cube-mapping algorithm

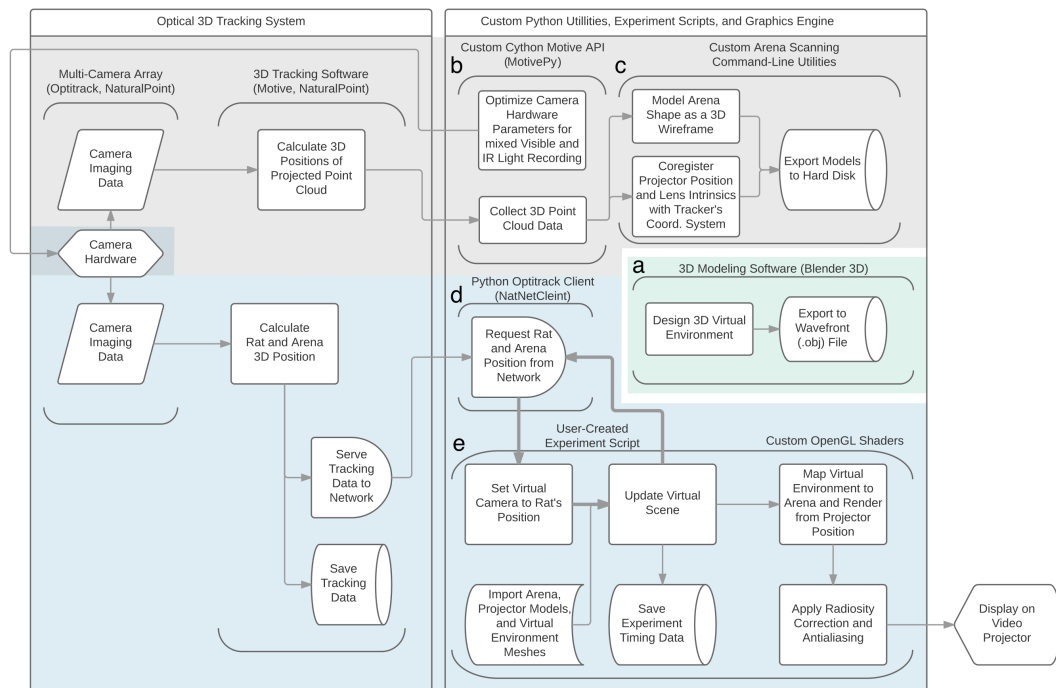


Figure 3.7: Ratcave VR hardware-software components flowchart. Each component, depicted as vertical parenthesis, takes information from one source and sends information to another source; information flow is depicted in direction of arrows. Detailed operations of each component are depicted as blocks, and software components are labeled by letter. (a) Blender 3D. The virtual environment is created before the experiment using 3D modeling software (right-center module) for loading into the VR experiment script. (Gray Zone) Tracking and Setup Coregistration. A Multi-camera array sends imaging data of the rodent's position on each camera to 3D tracking software, which combines the data from each camera image into a single 3D location and sends that position to the Ratcave VR environment (left, "Optical 3D Tracking System"). (b) MotivePy. The cameras' settings can be modified directly in a Python environment to make visible-light collection possible, a necessary step for arena scanning and projector calibration. (c) Ratcave VR_calibrate. Two command-line programs are used for arena scanning and projector calibration. The arena scanning program projects a moving grid of white dots on the arena surface, collects the 3D positions of the projected points via the camera array, and fits the resultant point cloud to a 3D mesh model of the arena. The projector calibration program maps single points displayed from the projector onto the 3D position of the arena, one at a time. It then uses OpenCV's camera_calibrate tool to use these mappings to find the position of the projector in the camera array's coordinate space. (Blue Zone) VR Engine. (d) NatNetClient. Rat position data is collected in real-time from the camera array and brought into the Python environment, for use in VR experiment scripts. (e) Ratcave VR. The virtual environment (VE) is rendered in a Python 3D graphics engine. The VE is loaded from file (created in Blender 3D), and on each display frame, using the rodent position data to move a virtual camera to the rodent's position in a virtual environment (blue zone). This process, encompassing the core of the VR engine, (get rodent position, move camera, update and render scene) occurs in a loop, repeated each frame, with the frames themselves sent to the GPU for arena mapping and shading (examples on Supp. Figure 1d) and then to the video projector (bottom-right corner). See the "Software" section in Online Methods for more details.

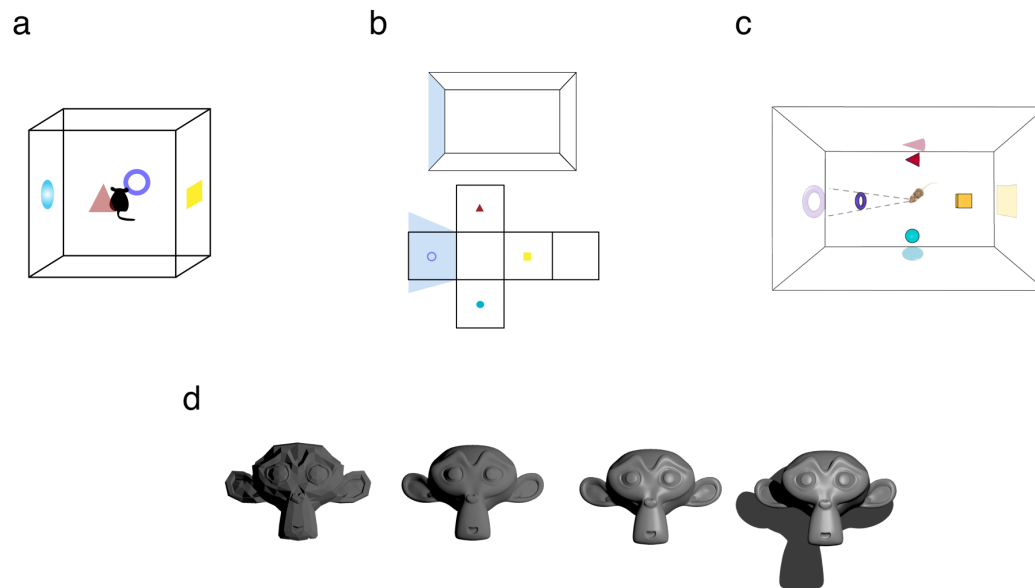


Figure 3.8: Cube-mapping and image rendering. (a-b) Schematic of the image-warping transformation of the rat's perspective view of the virtual environment to the projection on the arena. The image warping algorithm involves three steps: (a) The virtual world (consisting, in this example, of four colored 3D objects) is rendered 360 degrees about the rat's head position on the faces of the cube using a cube-mapping algorithm, (b) each wall's relative position to the rat is mapped to this 3D virtual world, and (c) all arena surfaces (walls and floor) are then warped from the perspective of a video projector mounted above the arena. This process is repeated every frame, maintaining the VR-rodent-arena despite movement of virtual objects, the rat, or the arena itself. (d) 3D lighting algorithms employed by Ratcave VR to increase spatial visual cues and visual richness of the virtual environment. Improvements are successively applied to the object, from left to right. First, diffuse reflections increase object brightness on parts of the object facing the light. Second, high-resolution objects are used, with smoothed surfaces, to further increase object detail. Third, specular reflections are added to provide subject-object-light triangulation cues. Finally, shadows are added to provide inter-object distance cues.

(Figure 3.8a-c), performs per-fragment lighting calculations (Figure 3.8d), and antialiases the resultant video output via custom OpenGL shaders (Supplementary Fig. 3.7). The resultant image is then projected onto the arena via the video projector.

Latency measurement

Motion-to-photon latency was explicitly measured using the following setup (Di Luca, 2010). A reference point, representing a VR observer, formed by a set of three retro-reflecting markers and a small LED, were attached to a bar that was rotated in the horizontal plain around a fixed point inside an arena by an AC motor and was tracked as described above. The VR system was programmed to generate a white spot that was offset in the horizontal plane from the reference point that followed a retroreflective reference marker. VR spot was thus rotating in the horizontal plain

following the rotation of the reference LED point. Both LED and VR spots were imaged using a high-speed-camera (Prime, Photometrics) at 250 Hz. The image stack was processed to detect both spots (Supplementary Fig. 3.9a) and temporal trajectories of X and Y coordinates of both reference and VR spots, which were analyzed to detect temporal offset between them using cross-correlation function (Supplementary Fig. 3.9b). The angular speed of rotation was varied between trials, and the resulting linear speed (tangential) was computed and used for latency-speed analysis (Supplementary Fig. 3.9c).

3.5.2 Animal Experiments Methods

All procedures complied with the European Communities Council Directive 2010-/63/EC and the German Law for Protection of Animals and were approved by the local authorities, following appropriate ethics review.

Subjects

Thirteen 6-month-old male Long-Evans rats (Charles-River, Germany) were used for the analysis of spontaneous exploratory behavior in virtual environments. Three of these thirteen rats were used for pilot versions of the experiments shown (Del Grosso et al., 2017), and ten were used in the final experiments, described in this paper. An additional male Long-Evans rat was used to record hippocampal neural activity in a virtual environment, as described in the "VE Shift Experiment" section. All rats were allowed ad libitum access to water and food, and were given a Froot-Loop cereal piece after each completed session (regardless of behavior during the session itself) in order to increase ease of handling. The ten rats used in the final behavioral experiments were pair-housed and kept on a reversed light cycle upon arrival to our animal room two weeks prior to handling, so that all experiments were conducted during their dark phase; all other rats were single-housed and kept on a normal light cycle. All rats were handled by the experimenter for five days prior to head-post implantation surgery and for two weeks post-surgery before exposing the rats to virtual stimuli in order to minimize the effects of subject stress on experimental results.

3.5.3 Behavioral experiments

We recorded the spontaneous behaviors of ten rats in four different virtual environments. Each session, conducted once per day, six days per week, over the course of two months, consisted of two phases: a block of three "visual cliff" experiment trials (one trial per experimental condition, described below in the "virtual cliff experiment" subsection), followed by one of three different ten-to-fifteen-minute arena exploration experiment sessions (described below in their respective experiment subsection), between which the rat was removed from the arena. Arena experiment

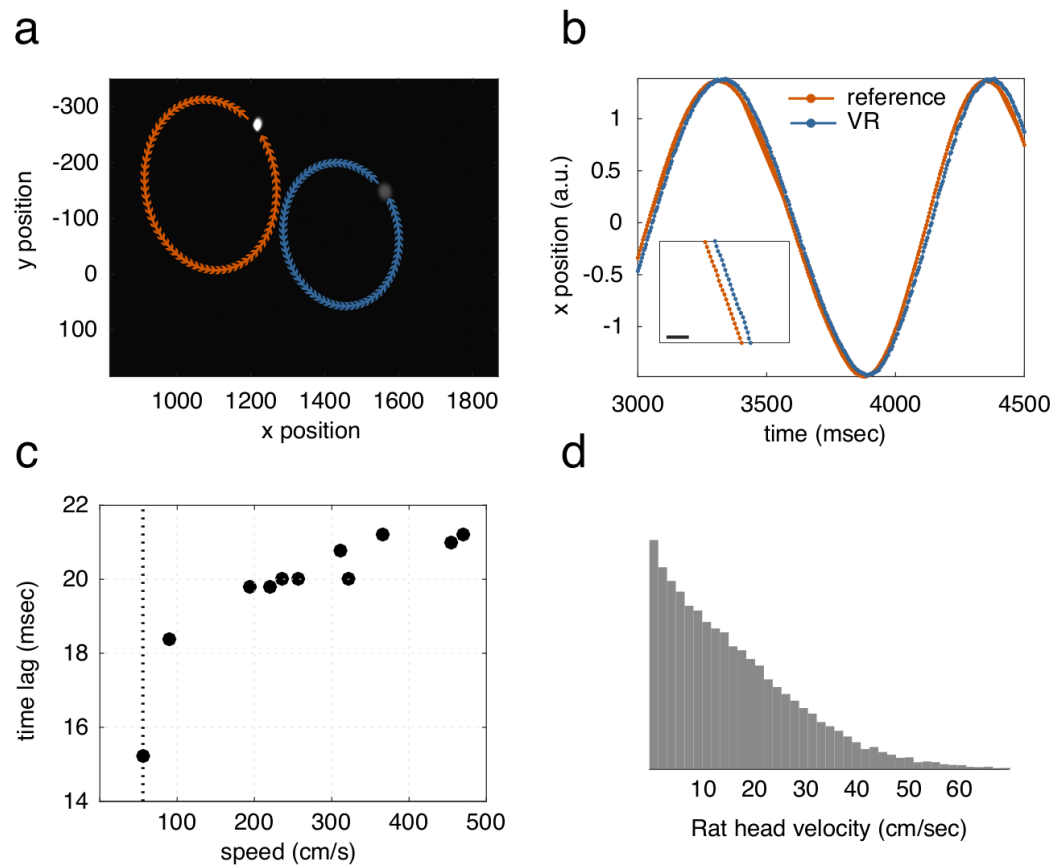


Figure 3.9: Motion-to-photon lag measurement of Ratcave VR system. (a) Raw image of the latency data collection procedure. A tracked object (reference point, left spot, orange) and its x-axis offset VR-represented projection (virtual point, right spot, blue) were recorded using high-speed camera as the reference point was rotated about a central point at different speeds. Arrows show the direction and shape of the reference and virtual points' trajectories. (b) Example of the time courses of the x-axis project of the reference (blue) and virtual (red) points. Inset, magnification of a section of time course. Note the delay between the two time courses, scale bar 10 msec. (c) Time lag between the reference and virtual points as a function of linear (tangential) velocity of the reference point motion. Note slow increase of time lag with speed of the reference point above 100 cm/s, with a minimum latency of 15 msec in the range of the head velocity of rats. (d) Distribution of head velocity in rats, note that all movement are contained within 60cm/sec (dotted line in c).

selection was done pseudorandomly using a Latin-square method, balancing condition presentation order between rats; the only exception to this is the novel-object experiment, which was conducted after the other experiments were completed. Rats were transferred from their home cage to the arena inside a small red-plastic-lined transport box, which the rats were trained to enter voluntarily to begin a session; this procedure allowed us to disorient the rats before each session by walking it throughout the recording room in the dark and place it in a random position in the arena, control timing and location of first-exposure to the virtual environment, and reduce subject stress through ease of handling.

Virtual cliff experiment

In each virtual cliff trial, the experimenter placed the rat on a 14 cm-wide board suspended 14 cm above the arena's floor, bisecting the arena into two halves. The two halves of the arena were assigned to one of two conditions: a "Cliff" side, in which the floor visually appeared to extend 1.5 meters below the actual arena floor level, or a "Safe" side, in which the visual texture of the floor matched the arena floor elevation (for an illustration, see Figure 2). Side assignment order was done pseudorandomly using a balanced Latin square method. In addition, rats were placed on a random side of the arena at the beginning of each trial to isolate the effects of turn-side bias; the transport box being locked in orientation parallel to the suspended board by magnets on both the board and box. After voluntarily exiting their transport box (described above), rats would visually explore the arena and suspended board, then jump down from the suspended board to the arena floor on one of two sides, after which the experimenter removed the rat from the arena. Each session lasted a maximum of 90 seconds, and rats that did not jump regularly within this time were removed from participation in the cliff experiment (three of ten rats). Cliff avoidance behavior was interpreted by observation of the rat jumping down from the board on the safe side. In order to isolate the effect of motion parallax on cliff avoidance behavior, each trial was done using one of three methods to present a cliff stimulus: either a "Real" cliff was presented as in the classic visual cliff paradigm, with a physical 1.4-meter cliff extension installed below a transparent nonreflective acrylic arena floor (Crescent Europe GmbH), a "Virtual" cliff was projected onto the plexiglass-covered opaque arena floor, simulating the checkerboard cliff walls as a dynamically-updated, geometrically-correct rendering of the real cliff pattern from the rat's perspective, or a "Static" cliff was presented by projected the same virtual cliff image but from only a single viewpoint (8cm above the center of the suspended board). Using the "Static" condition controlled for effects of spatial frequency and differences in geometry between the cliff and safe sides of the arena, allowing us to interpret cliff avoidance behaviors in the "Virtual" cliff condition in terms of the added motion parallax and geometrically-correct 3D aspects of the virtual cliff stimulus. Each of these three cliff experiment trials were performed each day for five

weeks (M=58 trials total per each of 7 rats, SD=8.5 trials per rat), with condition order pseudorandomly assigned using a balanced Latin square method.

Rat behavior was segmented based on head-tracking data into supported rearing on the arena walls and general exploratory behavior (Example trajectory shown in Figure 1c). Jumps were detected as trajectories that depart from the board and land on the floor. A rat's landing after jumping down from the board was detected based on the height of its head (threshold < 7 cm). Cliff avoidance behavior was interpreted by observation of the rat jumping down from the board on the safe side, after a period of visual exploration of the arena. Rats showed a preferential safe-side bias for both the Real cliff trials (M=66%, N=113, $p < .001$) and VR cliff trials (M=61%, N=114, $p < .05$), but not for the Static cliff trials (M=49%, N=110, $p = 0.92$, ns), tested for significance from chance (no bias, M=0.5) with a binomial test. Time between transport box exit and cliff side decision was highly variable (M=43.6 secs, SD=15.7 secs), and using two-sample Komogorov-Smirnoff tests, no significant differences were found between Static cliff trials (M=40.9 secs, SD=13.6 secs), VR cliff trials (M=44.5 secs, SD=13.6 secs), and Real cliff trials (M=45.6, SD=17.8 secs).

Virtual wall experiment

During the virtual wall sessions, rats were lowered into a random side of the arena after disorientation and allowed to freely explore the arena for 13 minutes, during which the experimenter left the recording chamber and monitored the rat remotely from an adjacent room. After one minute of free exploration in the arena without a projected visual stimulus, each session consisted of three four-minute experimental phases following an "ABA" experimental design (a pre-stimulus control phase, a virtual stimulus phase, and finally a post-stimulus control phase), each separated by a ten-second fade-to-black visual transition. All phases had a black-and-white checkerboard pattern projected on the floor and walls of the arena. During the stimulus phase, a virtual wall was presented so that it extended along the width of the arena at one of four positions spaced 20 cm apart along the arena's length. Wall position selection order was done pseudorandomly using a balanced Latin square method across subjects. The virtual wall texture matched the checkerboard pattern and contrast level to the arena surface, and the bottom edge of the wall hovered 1.5 cm above the floor of the arena in order to avoid a stationary floor line demarcating the virtual wall-floor intersection. Nine of the ten rats were included in this experiment (M=5.7 Sessions per rat, SD=1.16), with one rat excluded from full participation after he began repeatedly leaping onto the top edge of the arena (70 cm high), making behavioral analysis difficult.

Attraction to the virtual wall stimulus was measured by extracting several different metrics from the rat's movement trajectories during each session: occupancy near

the virtual wall, rearing behavior likelihood near the virtual wall, preferential rearing orientation facing the virtual wall, locomotion direction near the virtual wall, locomotion speed near the virtual wall, and locomotion trajectory shape perturbation near the virtual wall. Movement was classified as either "Rearing," "Locomoting," or "Stationary" in order to make metrics more context-specific. "Rearing" behavior was classified as time points in which the head was either positioned 15 cm above the ground or having a vertical velocity of 19 cm/sec. "Locomoting" behavior was classified as time points in which the rat was both not rearing and had a horizontal velocity greater than 1cm/sec, smoothed with a 500-msec hamming window. Real-wall thigmotaxis behavior (Locomotion or Stationary behaviors within 5cm of the arena walls) was also recorded and used to exclude data for many metrics.

The proportion of time spent near the virtual wall position (<15 cm) was measured for each of the three experimental phases (Pre-VR, VR, and Post-VR). Rearing and Thigmotaxis behaviors were excluded from this analysis. When the virtual wall was present, rats occupied the arena within 10 cm of the virtual wall's position for a greater amount of time in the virtual wall phase (Median=45.1 secs, SD=26.2 secs) than would be expected by chance, as compared to the other three potential wall positions (Median=55.6 secs, SD=23.8 secs), $Z=6.96$, $p<.001$, $N=55$ Sessions. No increased virtual wall position occupancy was found for the Pre-Stimulus phase (Wall: 23.9 ± 27.4 secs, Rest: 79.5 ± 27.4 secs, $Z=0.59$, $p=.55$, $N=57$ Sessions, ns) nor for the Post-Stimulus phase (Wall: 12.9 ± 16.3 secs, Rest: 53.6 ± 26.5 secs, $Z=-1.6$, $p=.11$, $N=55$ Sessions, ns). This occupancy was localized, with rats spending a greater proportion of time within 5cm of the wall than the surrounding 10cm on either side for the Stimulus phase ($N=56$ Sessions, Median=48.9%, SD=13.7%, $Z=4.67$, $p<.001$) than the Pre-Stimulus phase ($N=54$ Sessions, Median=22.7%, SD=13.2, $Z=0.23$, $p=.82$, ns) or Post-Stimulus phase ($N=50$ Sessions, Median=29.2%, SD=12.7%, $Z=-0.5$, $p=.13$, ns), Figure #d. Rearing behaviors were also localized within 5cm of the virtual wall vs the 15cm surround (Pre-Stimulus: $33\% \pm 24\%$, $Z=0.91$, $p=.37$, ns; Stimulus: $59\% \pm 19\%$, $Z=6.08$, $p<.001$; Post-Stimulus: $38\% \pm 26\%$, $Z=0.69$, $p=.49$, ns), $K=37.8$, $p<.001$, Figure #d.

While the rats generally locomoted less often and more slowly over the course of a session, showing a negative linear correlation ($r=-.0083$, $p<.001$, GLM), their median locomotion speed near the virtual wall's position also differed by phase ($K=12.7$, $p=.0018$). This speed was lower for the virtual wall phase (12.8 ± 3.7 cm/sec) than for the pre-stimulus phase (18.52 ± 6.8 cm/sec, $Z=3.8$, $p<.001$), although not for the post-stimulus phase (13.3 ± 11.9 cm/sec, $Z=-1.0$, $p=.32$). No significant difference in near-virtual wall locomotion speed was found between the pre- and post-stimulus phases ($Z=0.26$, $p=.79$). Thigmotaxis-like locomotion orienting along the virtual wall axis, defined as locomotion in a direction within 45-degrees of the short axis of the arena within 5 cm of the virtual wall, was also observed to differ between phases, $K=59.8$, $p<.001$. This locomotion orientation was more likely during the the

virtual wall phase ($56\% \pm 9\%$) than the pre-stimulus phase ($42\% \pm 12\%$, $Z=-6.8$, $p<.001$) and post-stimulus phase ($38\% \pm 19\%$, $Z=4.96$, $p<.001$). No difference in short-axis locomotion likelihood was observed between the pre- and post-stimulus phases ($Z=0.48$, $p=.63$, ns), Figure #f.

Although we did not collect enough trajectories per session to compare them on a per session basis, we also compared trajectories through the virtual wall's position by phase and found that these trajectories were more likely to change direction (d -Theta > 5 deg.) during some phases than others ($X^2=7.02$, $p<.03$, chi-square test). Probing tests found that this increase occurred during the virtual wall phase (65% , $N=106$), as compared to the pre-stimulus phase (49% , $N=199$), fisher=1.9, $p = .01$. No difference was found between virtual wall phase and post-stimulus phase trajectories (54% , $N=87$, fisher=0.6, $p=.14$), nor between the pre- or post-stimulus phases (fisher=1.2 $p=.52$). Straight trajectories passing through the virtual wall's position ($N=214$) also differed by speed depending on the phase, $K=30.7$, $p < .001$. This increase was from slower trajectories during the virtual wall phase (Median=23.31 m/sec, $SD=11.38$, $N=69$) than the pre-stimulus phase (Median=39.6 m/sec, $SD=13.92$, $N=98$, $K=.49$, $p < .001$) than the post-stimulus phase (Median=42.7 m/sec, $SD=15.3$, $N=47$, $K=.63$, $p < .001$). No differences were found between pre-stimulus and post-stimulus wall-crossing trajectory speeds, $K=.25$, $p=.06$.

Virtual object exploration

As in the Virtual Wall experiment, rats were lowered into a random side of the arena after disorientation and allowed to freely explore the arena for 13 minutes, during which the experimenter left the recording chamber. The same nine of ten rats were included this experiment as in the virtual wall experiment. This experiment also employed the same experimental design (ABA sequence, four minutes per phase) as in the wall experiment. In these sessions, two enclosed glass cylinders (24 cm high, 10 cm in diameter) were placed 40 cm apart along the center of the arena. An object (6 cm in diameter, its lowest point 2 cm above the arena floor) was presented inside one pseudorandomly-selected cylinder: either a white 3D-printed plastic geometric shape (seven unique shapes were used) by automatically raising the object into the cylinder through a hole in the arena floor, or a matching virtual object was presented inside the cylinder by projecting it onto the arena wall, so that the cylinder was always between the rat and the virtual projection (for example of stimulus, see Supplementary Video 3.7). Experimenters checked for attraction to virtual and real objects by measuring preferential exploration between the two cylinders during the stimulus presentation phase, comparing this exploration bias to the pre-stimulus phase to factor out general arena side preference. Exploration was defined as periods of rat head occupancy within 12 cm of the center of the dome, head orientation pointing within 90 degrees to the same point, and non-rearing, non dome-mounting behavior. s Attraction to a virtual object was measured by comparing

the proportion of glass dome exploration (i.e. left vs. right) between phases: the pre-stimulus, stimulus (in which an object contains a virtual or real object), and post-stimulus phases. Exploration was defined as periods of rat head occupancy within 12 cm of the center of the dome, head orientation pointing within 90 degrees to the same point, and non-rearing, non-dome-mounting behavior. Two-sided, Wilcoxon paired-sample tests were used to infer exploration bias between the two objects. In sessions where a real object was presented (N=38), rats spent a larger proportion of time exploring the real-object-containing glass than the empty glass during the stimulus phase (M=60%, SD=18.5%, Z=2.81, p=??), but not in the pre-stimulus phase (M=46%, SD=15.4%, Z=-1.34, p=.16, ns) nor the post-stimulus phase (M=58.5%, SD=18.5%, Z=1.75, p=.08, ns). In sessions where a virtual object was presented (N=45), rats explored the virtual-object-containing glass for a larger proportion than the empty glass during the stimulus phase (M=57.5%, SD=20.1%, Z=2.39, p=.02), but not during the pre-stimulus phase (M=50%, SD=13.4%, Z=-.13, p=.9, ns), nor the post-stimulus phase (M=41.5%, SD=30.2%, Z=-1.74, p=.08, ns).

Virtual spatial novelty detection experiment

As in the virtual wall experiment, rats were lowered into a random side of the arena after disorientation and allowed to freely explore the arena, during which the experimenter left the recording chamber. The same nine of ten rats were included this experiment as in the virtual wall experiment. Each 15-minute session consisted of the three same phases as the other experiments (a 4-minute Pre-Stimulus phase, a 4-minute Stimulus phase, and a 3-minute Post-Stimulus phases, with two objects presented instead of one), plus an additional 4-minute Stimulus phase, in which the same two objects as the previous phase were presented, with one of them in a different arena location; because we were interested in differences in exploration durations between the two objects between phases, the first stimulus phase is termed the Familiar Stimulus Position phase and the second the Novel Stimulus Position phase. The "fixed" and "movable" objects were presented in different sets of locations in the arena, in order to maximize and control the inter-object distance between the Familiar and Novel Stimulus Position phases and to increase the object distance from the arena walls, with Fixed objects appearing at one of three central locations, and Movable objects appearing at two of the four arena corners (Figure 3.10). To increase disparity between the Familiar and Novel positions, the Novel position within a session was always set to the opposite end (length-wise) of the arena, on a randomly-selected corner. To minimize clipping from the top edge of the arena walls, the 5cm-diameter objects' maximum height was set to 7.5cm off the arena floor, and the near-clipping distance was set to 2.5 cm from the rat's eye position. Object exploration behaviors were defined as locomotion trajectories within 5 cm of the object's center. Occupancy densities between object and non-object inner-arena locomotion trajectories were calculated by dividing object occupancy times by non-object-intersecting, same-diameter, randomly-sampled (N=400 samples per phase)

locations in the arena ("Sham" objects), in an area greater than 15cm from the arena walls to avoid thigmotaxis behaviors (Figure 3.10d). Spatial Novelty-evoked exploration preference was calculated by comparing the ratio of occupancy in the Fixed and Movable objects between the Familiar and Novel Stimulus Position phases (Figure 3.10g).

In this experiment, rats had the ability to run through the virtual objects (as opposed to the Virtual Object Exploration experiment, in which a glass dome blocked tactile interactions with the virtual objects); as a result, exploration was defined as locomotion within 5 cm of a virtual object. To define general attraction to the virtual objects, the density object exploration for both objects was compared to locomotion in non-object, non-thigmotaxic areas of the arena (detailed in the Methods section above) in the Familiar Stimulus Position phase, using a two-sided, Wilcoxon paired-sample on the mean object-sham exploration density for each phase. For this analysis, sessions from the N total recorded sessions if the rat did not enter at least one of the two arena regions. A significant increase in exploration in the object area over the sham area exploration density was found between the Pre-Stimulus and Familiar Stimulus phases (N=## Sessions, $Z=2.36$, $p=.018$); this object exploration preference over the sham area was found for the Familiar Stimulus phase (M=65.5%, SD=45%, $Z=2.75$, $p=.005$), but not for the Pre-Stimulus phase (M=49%, SD=45%, $Z=-1.38$, $p=.17$, ns), Figure #d. Both the Post-Stimulus and Novel Position phases were excluded from the object exploration analysis; the former for inadequate sampling reasons (phase duration was one-third that of the other sessions) and the latter to avoid memory-based confounds with the familiar object position. Object exploration was also inferred by a decrease in locomotion speed for trajectories passing within 5 cm of the object locations. This measurement was only done for the Pre-Stimulus Familiar phase and for the centrally-located Fixed object location, in order to decrease memory confounds and artifacts from trajectory changes attributed to the arena walls. A significant decrease in locomotion speed was found in the phase in which the virtual object was present (N=48 Sessions, M=##, SD=##, $Z=-2.45$, $p=.01$), with no significant difference in object-crossing trajectory speed during the Pre-Stimulus phase (N=27 Sessions, M=##, SD=##, $Z=-.26$, $p=.79$, ns). Spatial novelty detection was measured as an increased bias in object exploration toward the Movable object over the Fixed object position between the Familiar and Novel Stimulus Position phases, assessed with a two-sided Wilcoxon paired-sample test. This novel object position exploration preference was found when comparing the Novel and Familiar Stimulus Position phases (N=42 Sessions, $Z=2.67$, $p=.028$); this difference is found as a significantly-higher Movable vs Fixed object exploration in the Novel Stimulus Position phase (M=65%, SD=##, $Z=2.66$, $p=.008$), with no significant difference found for the Familiar Stimulus Position phase (M=51.5%, SD=##, $Z=0.45$, $p=0.65$, ns).

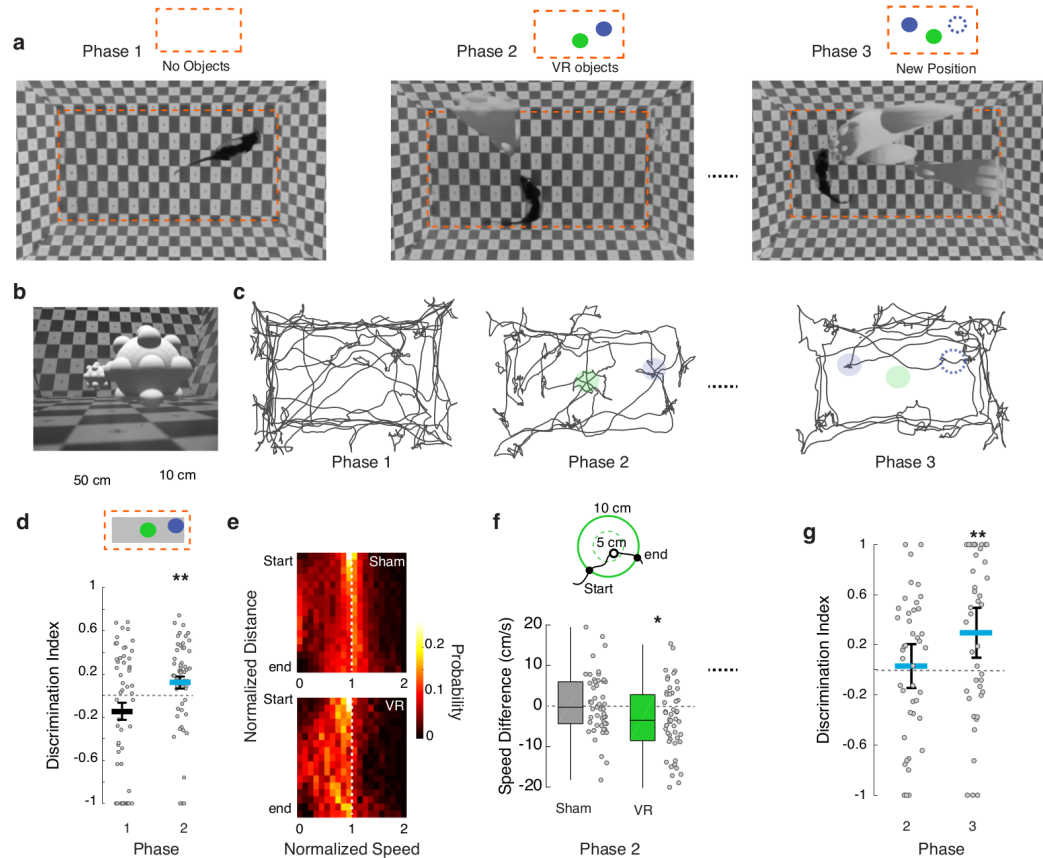


Figure 3.10: Spatial Novelty Exploration Experiment. (a) Experimental protocol. Each session consisted of a Pre-Stimulus, Familiar Stimulus Position, and Novel Stimulus Position phase. Of the two objects shown in the Stimulus phases, one was Fixed in position between phases (green), and the other moved to a different arena corner (blue) (b) Image of virtual objects from rat's perspective. (c) Example rat trajectories from a single session for each phase. (d) Mean session discrimination index of near-object (< 5cm) locomotion vs far-object (>5cm, shaded area above) locomotion for the Pre-Stimulus and Familiar Stimulus phases; object exploration significantly increased in the latter phase. a single object at the center of arena (green color in a). (e) Locomotion trajectory speed decreases as rats approached the center object. (f) Top: Schema showing near-object locomotion trajectory thresholds; only trajectories passing within 5cm of object center were included. Bottom: box plot of the speed difference between starting position (10cm from object) to closest-object point in trajectory. (g) Discrimination index of the VR object pair between the Familiar and Novel Stimulus Position phases. Movable object location exploration (positive direction) increased during the Novel Stimulus Position phase. Error bars represent 68% confidence intervals. Asterisks represent p values below different alpha thresholds (*: $p < 0.05$; **: $p < 0.001$).

Virtual environment shift experiment

During the VE shift experiments, the rat was allowed to explore the arena for 40 minutes per day, broken up into 10-minute phases, between which the rat was removed from the arena by the experimenter. This experiment had two conditions, which were alternated: a "Normal" condition in which a static checkerboard projection illuminated the arena walls and a "Shifted" condition in which a virtual arena of matching dimensions but spatially shifted 20 cm along the arena's length was projected. This effectively resulted in the shift of one virtual wall to the outside of the arena and another to the inside of the arena, producing an effect similar to the virtual wall experiment. The rat was exposed to the arena and these two conditions twice for the first time on Day 1 (Normal1, Shift1, Normal2, Shift2 in Fig. 3.5b-c) and then repeatedly to the same conditions, as well as other VR manipulations in several sessions on Days 2 and 3 (data not shown). On Day 4 two sessions were recorded under Normal and Shifted conditions and an extra session under condition "Star field" was introduced (Normal, Shift and Star field in Fig. 3.5d-e). This condition consisted a 3D grid of repeating white cubes, which extended 1 meter beyond the walls and floor of the arena.

Surgery and electrophysiological recordings

Rats were anesthetized with a three-component mixture (Fentanyl .005mg/kg, Midazolam 2mg/kg, Medetomidine .15mg/kg); this compound also provided analgesia for the first part of the procedure. A 1.5% concentration of isoflurane in oxygen was used to maintain depth of anesthesia for the rest of the surgery. In animals used for behavioral assays, four small screws were fixed into the skull to provide support for our head post. In one rat, a silicon probe (NeuroNexus, Buzsaki 32 design, 4 shanks, 8 sites 25um vertically spaced) was implanted following procedures described elsewhere Sirota 2008. Briefly, a cranial window of 2 mm² was opened, centered on the following coordinates from bregma: -3.36 mm AP and +2.6 mm ML. The silicon probe, mounted on a custom-made microdrive, was inserted in the center of the craniotomy with the shanks aligned parallel to the septo-temporal axis of the hippocampus (45 degrees parasagittal). The probe was lowered to a distance 1mm from the surface, and the drive was affixed to the skull. After the rat recovered from surgery (1 week), the probe was lowered 50-150 microns daily until we observed the typical profile of activity of CA1/CA2 cell layer; namely, spiking activity and ripple oscillation signal in the LFP. Histological verification of the location of the recording electrodes was done after the conclusion of the experiments 3.11.

Data acquisition and processing

Extracellular signals were amplified and filtered by multi-channel preamplifiers (Plexon, 20x, 1-5000 Hz). Wide-band extracellular and intracellular signals were digitized at a 20 kHz sampling rate with 16-bit resolution and stored for offline analysis using

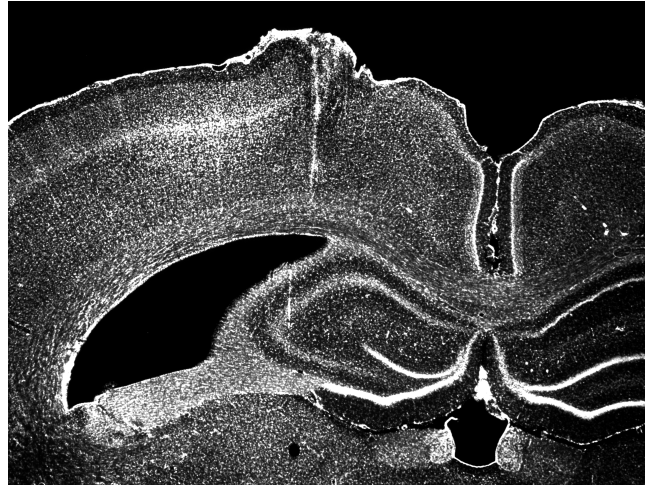


Figure 3.11: Histological stained slice verification of electrode implantation location in hippocampus

a multichannel acquisition system (DigiLynx, Neuralynx). Raw data were preprocessed using a custom-developed suite of programs (neurosuite.sourceforge.net). The wide-band signal was downsampled to 1.25 kHz and used as the local field potential signal. For spike detection, the wide-band signal was high-pass filtered (>0.8 kHz). Single units were isolated semi-automatically by an open-source spike-sorting program KlustaKwik (<http://klustawik.sourceforge.net>) Harris 2000 and refined manually using open-source GUI software (<http://klusters.sourceforge.net>; <http://neuroscope.sourceforge.net>) Hazan 2006. The quality of isolated single units was confirmed by an isolation distance metric and a clean refractory period.

3.5.4 Data Analysis

All data analysis was performed using custom-written code in Python and Matlab (Mathworks, Inc.).

Data representation and statistics

Sample size was selected after an a priori power analysis for each experiment, based on effect sizes obtained from the literature and from pilot studies. Pseudo-randomization techniques were used to balance rat cage order and experimental parameters as described in the description for each experiment. Blinding was not performed during data collection, since the experimenter was not present in the recording chamber during most experiments (or, in the case of the cliff experiment, did not handle the rat directly in order to keep from influencing their decision) nor analysis, although automated methods for experimental condition identification and behavioral classification were used to ensure consistency in data processing between recording sessions.

Behavioral state classification

Behavioral state of the rat was classified based on the speed and height of the head. Using data-derived thresholds for these variables, we defined running (speed > 3cm/s & height < 15.4cm), immobility (speed \leq 3cm/s & height < 15.4cm) and rearing (speed \leq 3cm/s & height > 15.4cm).

Electrophysiology Data Analysis

Brain state segmentation

Hippocampal activity was segmented into two states: theta and non-theta. An HMM Gaussian mixture model based on the hippocampal CA1 pyramidal layer spectral power ratio between the 6-12 Hz band and the sum of the 1-5 Hz and 15-18 Hz bands of the whitened LFP was used to separate theta and non-theta states. All further analysis of the hippocampal place cells was constrained to theta-associated periods.

Place cells analysis

Only hippocampal pyramidal cells with place fields that were active in the arena were included in the analysis. Spike width and firing rate were used to separate pyramidal cells from interneurons. In the sessions used in this paper, 309 of 367 cells were classified as putative pyramidal cells (Day 1: 166 of 168 cells; Day 4: 143 of 182 cells). Place cells were defined as putative pyramidal units with a place field peak firing rate of at least 3 Hz, having less than three spatially-separated firing rate peaks in all trials, and maintaining a stable spike waveshape across all sessions of the day. After filtering based on these selection criteria, 39 total pyramidal cells for the two days (20 and 19 cells, respectively) remained.

Place fields were calculated based on a k-nearest neighbor algorithm, which selected for periods in which the speed of the rat's head was greater than 5 cm/s and intersected with periods of theta oscillation state. The k-nearest neighbor estimate of the mean firing rate was calculated given the position of the rats head and each unit's smoothed firing rate. The unit firing rate was smoothed using a 800 ms rectangular window, convolved with the time-resolved spike histogram and downsampled to 30 Hz. The maze was binned with 2 cm square bins. For each bin, the smoothed unit firing rate was sorted by its distance to the bin center. The first 300 nearest-neighbor time bins were collected and averaged to derive the mean rate of that bin. Bins with less than 300 neighbors within a radius of 12.5cm were assigned to be empty. This procedure provides a data-adaptive and robust estimate of the spatial rate map in contrast to conventional estimation methods (ratio between spatially smoothed spike count and rat occupancy maps). Qualitatively, though, both measures gave the same results.

The procedure used for place field map estimation has additional benefits, as it allows robust estimation of the parameters of the place field based on the bootstrap procedure. The variance of the place field center was estimated by bootstrapping each unit's 1000 random subsets of two-second chunks of the rodent's trajectories (75% of total Trial time). The place field center at each iteration was calculated by thresholding the rate map by the firing rate at the 95 percentile of all iterations. All bins above the threshold were assigned a 1, and all other values were assigned a 0. The above-threshold bins were segmented using Matlab's `bwboundaries` function into spatially contiguous patches, each of which represented a place field. The area, the rate-weighted center of mass, and the maximum and minimum firing rate were calculated for each patch. Only the main (largest and highest firing rate) field was used for further analysis. The location of the peak rate within the patch was computed for each bootstrap sample, and the resultant mean estimate was used as an unbiased estimate of the x-y position of the center of the place field and used for the further analysis of place field remapping. To quantify the effect of the VE shift on the place fields of the active population of place cells, we computed the displacement of the place field center between consecutive sessions (Normal to Shift, Shift to Normal etc). The Kruskal-Wallis test was used to find an overall difference in population means between sessions along each axis of the arena, and significant axes were probed for individual differences between sessions using a Wilcoxon paired-rank test with p-values corrected for multiple comparisons using Benjamini/Hochberg False Discovery Rate method.

Chapter 4

Discussion

Introducing a freely-moving CAVE system into a rodent electrophysiology lab brings many benefits towards research in perception and cognition: better control over sensory environments, observation of naturalistic behaviors, control over sensorimotor information flow, and artificial 3D environment topology stimuli. When combined with functional imaging of neurophysiology during these tasks, many discoveries in cognitive science are made possible. While any video display showing a 3D environment set in closed-loop with a motion tracking system can technically be called a VR system, the primary yardstick in quality for a VR system should be its potential for both increasing immersion and reducing immersion-breaking moments. Immersion into a virtual environment requires wide-field, low-latency displays, natural and intuitive interaction mechanisms, and high levels of intersensory coherence between motor behaviors and the visual, vestibular, and somatosensory systems. The ratCAVE system meets these requirements by avoiding vestibulomotor coherence artifacts or reduced motor affordance usually introduced by locomotion simulation; instead, rodents remain as freely-moving and unrestrained as in real-world situations, with a high-speed, always-360-degree projection display to keep up with their behavior. Through VR experiments with untrained, spontaneously-behaving rats, we showed that rats treat virtual environments in a manner consistent with real-world stimuli and represent virtual objects, distances, boundaries, and environments as three-dimensional aspects of their real-world environment. Freely-moving CAVE setups thereby provide experimental benefits for neuroscientists as an augmented-reality solution.

4.1 Development of the ratCAVE VR Setup

Developing the ratCAVE VR setup for our neuroscience research involved an incremental design process for each component: the 3D graphics engine, the arena, and head mount designs, and the arena reconstruction and projector calibration algorithms. This section describes intermediate solutions for each component and provides detailed information on the algorithms used in the calibration procedures for the VR setup.

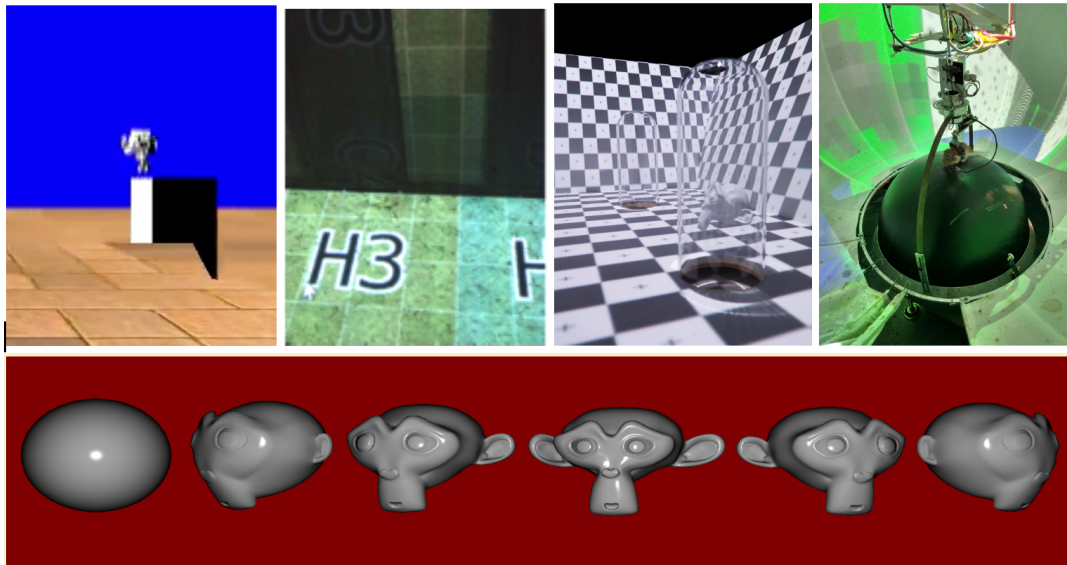


Figure 4.1: Development of the ratCAVE 3D graphics engine. A) Blender3D rendering of the first proof-of concept model demonstrating real-time dynamic projection mapping. B) First real-world projection mapping onto the recording chamber walls, this time with the first iteration of the VR software, then-titled “HippoVR”. C) Final projection onto the arena by the RatcaveVR software. D) The Ratcave graphics package in use by a treadmill rodent VR setup. E) Dynamic mesh warping demonstration in Ratcave, a feature useful for some psychology research.

4.1.1 The 3D Graphics Software

Early prototypes were simulated and rendered to simple projection mapping models using Blender3D (Figure 4.1a-b), an open-source 3D graphics program that supports Python scripting. This software does not support cubemapping, though, and so the Ratcave 3D graphics Python package was developed. This graphics package was found to be useful as a standalone library that could be used not only in our VR setup (Figure 4.1c), but also in another lab’s treadmill-based rodent VR setups (Figure 4.1d) and as an extension to existing psychology stimulus-presentation software in Python (Figure 4.1e).

4.1.2 The Arena

While the ratCAVE VR setup approach does allow projection onto any concave arena, it was helpful to work with a single arena frame that could flexibly accommodate multiple experiments. A four-sided rectangular arena was chosen for its one-to-one match with the aspect ratio of the video projector, with sloping walls to decrease projection distortion and increase image resolution and brightness on the projection surfaces, which is also why we chose a front-projection strategy (Figure 4.2). The simplicity of the design also allowed the 3D model to be hand-modeled in the earliest stages of the project before arena scanning was implemented. The projection surface was coated with a white extra-matte lacquer in order to minimize

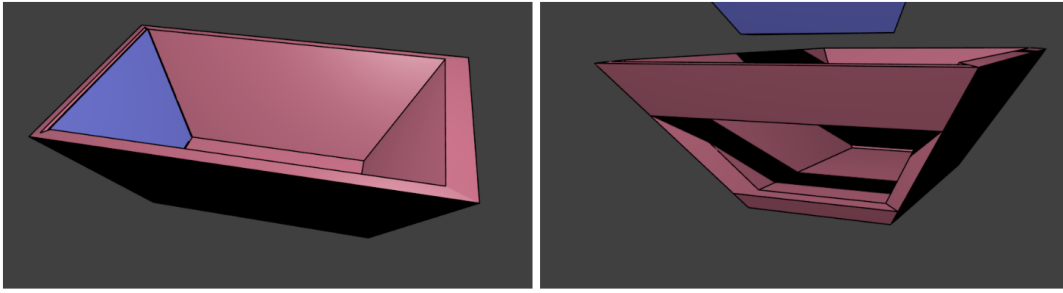


Figure 4.2: The ratCAVE arena design. Arena shown here with false color to highlight the arena's removable wall.

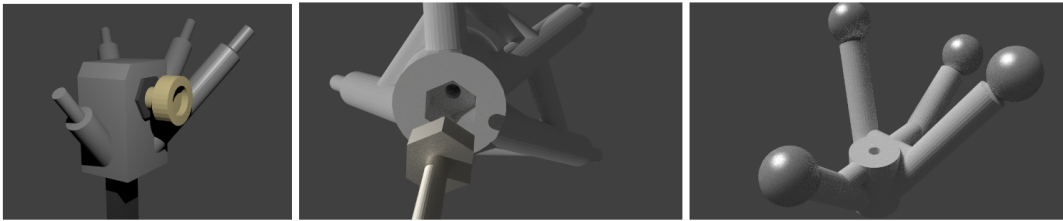


Figure 4.3: Early designs for the trackable rodent head mount, a 3D-printed rigid body for passive, retroreflective spherical markers.

moving bright spots from specular reflections and selected for its scratch-resistant, cleanable properties, useful during rodent handling. Multiple, insertable floors of different sizes (100cm x 50cm & 120cm x 70cm) were also created for the arena; by sliding them in and out of the arena's bottom through a removable wall (blue in Figure 4.2), they could be rapidly replaced, cleaned or even modified by adding embedded water ports for operant conditioning training or transparent glass domes for our object attraction experiment.

4.1.3 The Head Mount

The trackable head mount was conceived as a 3D-printed skeleton that could keep passive tracking markers coated with retroreflective beads in a rigid body, enabling easy rat tracking and throughout the duration of an experiment (Figure 4.3). Head mounts were fixed in orientation to the head post by fitting either to a flat bar or secured nut, and because they were secured by a magnet or hand screw, they could be quickly removed and re-secured between experimental sessions without concern about inter-session placement differences. In addition, because the same mounts could be placed on any rat, the tracking system software did not need calibration for each individual subject. Over time, the head mount became lighter and went from 5 markers to only 3 (the minimum required for 3D orientation tracking), as we found that marker occlusion in the small skeleton volume produced extra tracking noise that compromised the benefits of the extra tracking markers.

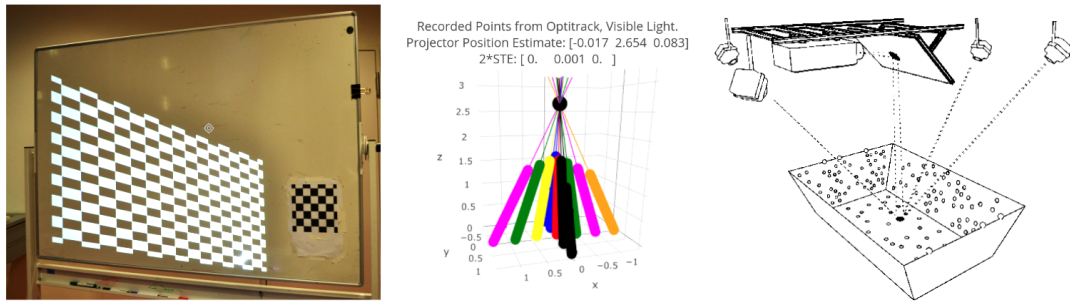


Figure 4.4: Development Progression of an automated, reliable, and flexible projector calibration method for the ratCAVE setup. A) Projector intrinsic parameter calibration via projector-camera calibration on a checkerboard pattern. B) A manually-intensive ray-Intersection method using the tracking system to compute all parameters. C) The final, automated, OpenCV-based point-cloud method to estimate all parameters.

4.1.4 The Projector Calibration Method

Accurately estimating the video projector’s position and rotation in the tracking space (called the projector’s “extrinsic” parameters) as well as its lens characteristics (its “intrinsic” parameters) is necessary for accurately projecting the virtual environment onto the projection surface.

Early attempts were done to estimate the intrinsic and extrinsic parameters separately using a fiducial imaging approach (Figure 4.4a), and later the tracking system itself was used to estimate all parameters simultaneously. At first, this was done by finding one-to-one correlations between individual light rays extending from the video projector (collected by physically moving a piece of paper along a ray) and finding the intersection and dispersion of the rays to infer the projector’s parameters (Figure 4.4b). The final solution was to use OpenCV’s camera calibration module, which finds the parameters given a set of 2D and 3D points projected on the arena surface (Figure reffig:calibrationc). The nonlinearity of the arena’s surface was key here for reducing the 3D-2D projection equation to a unique solution. This technique allows us to perform full tracker-projector coregistration in an automated manner.

4.1.5 The Arena Mesh Reconstruction Method

In order to accurately position the projection of the 3D scene on to the arena, it is also necessary to know the shape and position of the arena surfaces. The final solution here was similar to that of projector calibration, where we used the 3D tracking system as a kind of 3D scanner for arena mesh reconstruction. To do this, thousands of light points were projected onto the arena and their positions stored as a 3D point cloud (Figure 4.5a). Each of these points was then identified as belonging to a single wall or floor of the arena (Figure 4.5b) and the intersections between each arena plane was found to form the vertices for the resulting mesh (Figure 4.5c). To do this, the algorithm used for ratCAVE relies on the assumption that the walls are flat,

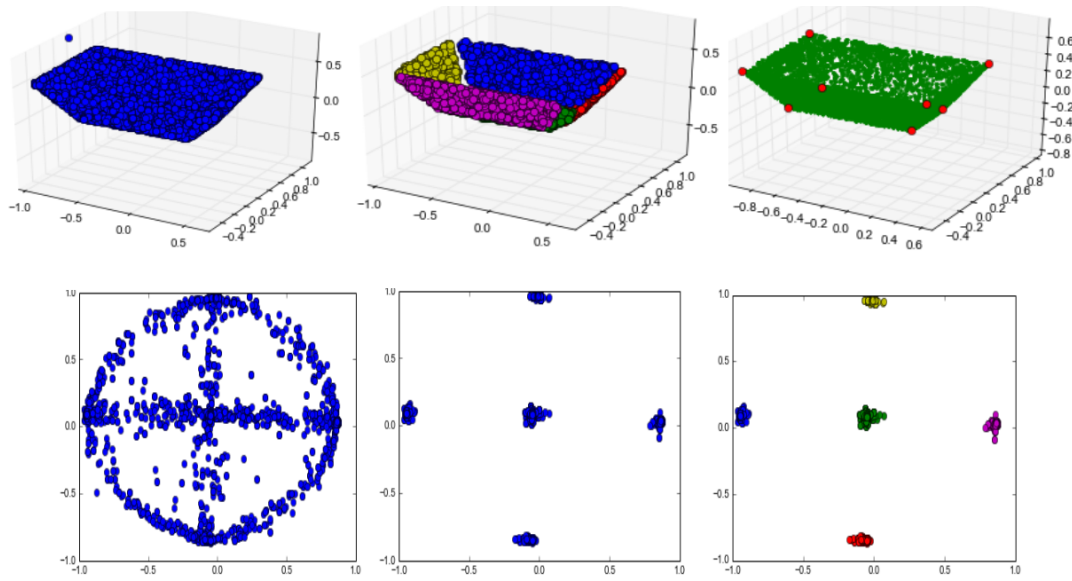


Figure 4.5: Arena Mesh Reconstruction Algorithm. A-C) Overview of the algorithm. A 3D point cloud collected by tracking points projected onto the arena was classified as belonging to a given surface and the intersections between each surface found in order to build a 3D mesh. D-F) The surface clustering algorithm. Normals of each point's local neighborhood were filtered by length of the normal, clustered, and labeled using the K-Nearest Neighbors, Principal Component Analysis, and Gaussian Mixture Model methods.

which holds true for our arena. Each point is associated with its K nearest neighbors in the point cloud and the orthogonal direction of each point's local neighborhood (the normal to the plane) inferred through Principal Component Analysis (PCA) dimension reduction (Figure 4.5d). Points located near the walls and corners of the arena were filtered out of this collection by low-pass thresholding the length of the normal vector, which is longer (contains higher amount of variance for the cloud) for non-flat distributions (Figure 4.5e), and n unsupervised clustering method (Gaussian Mixture Modeling) was used to label each cluster, and thereby the arena surface (Figure 4.5f). A final PCA on each face's point cloud provides the direction and intercept of each wall, and the linear equation for each wall's intersections are solved to provide the arena's corner vertices, the final component for the arena 3D Mesh reconstruction.

Further developments are planned for the system, including improved interfaces for designing and presenting experiments, 3D glasses for the rodents to study depth perception via stereopsis, and a larger, lighter arena than the current wooden one, where more-varied experiments can be conducted and vestibular stimulation via arena movement can be introduced. Overall, however, the current system has proven useful for several experiments, and the iterative design process has helped in that regard.

4.2 Increasing VR Immersion: Further Directions

While the ratCAVE setup incorporates many strategies to both provide a rich sensory environment and reduce sensory cue conflict, many cue conflicts remain to be addressed. Binocular depth cue via stereopsis, for example, remains to be incorporated into our rodent environments, and vergence-accommodation conflict (VAC), a visual depth conflict related to accommodation of the retina is a source of eye strain on VR subjects and requires novel display technologies to resolve in all VR setups (Kramida and Varshney, 2015; Huang, Luebke, and Wetzstein, 2015). Implementation of stereopsis in animal VR systems has already been successfully applied in a VR system for praying mantises, where the mantis used binocular depth judgments while wearing blue-green anaglyph filters over its eyes to more accurately hunt virtual prey (Nityananda et al., 2016). While they all have the potential to help evaluate a scene, actual utilization of these depth cues is both situation-dependent (Schrater, 2000) and subject-dependent (Westerman and Cribbin, 1998). While this cue is undoubtedly valuable for humans, the utilization of binocular disparity cues for depth perception by rodents is still poorly understood. The interactivity elements of the setup can also be vastly improved; while the experiments demonstrate here that the visual sensory environment is sufficiently immersive for some level of immersion, adding the ability for the rodents to interact naturally with virtual objects would make a big difference. The setup's tracker-based dynamic projection mapping feature could help with this by rendering onto real-world moving objects that the rodent can physically touch and even move during an experiment.

The experimental design factors that affect immersion in virtual environments also remains to be studied. For example, experiments that manipulate visuospatial cues to change a subject's perceived location may be biased or suppressed by the strength of subjects' memories of the spatial relationship between virtual environment and real-world environments, disorientation in darkness before moving to a virtual environment may be necessary as part of the experiment's protocol, as is done in Morris water maze studies. This precaution may also increase the variance of spatial coding in physiological studies (Stackman et al., 2003) and decrease the correlation between the spatial representation of subsequent environments (as found in Taube and Burton 1995). While disorientation is simple to perform in freely-moving CAVE setups because the VR arena has the same physical properties as a real-world arena, VR setups that are fixed in place would need to find an alternate solution.

The flexibility of visual scene manipulation in virtual environments also allows for refined research in the effects of landmark stability on spatial representation. This is a complex field of study; for example, landmarks that have a history of moving have a weaker role on head direction cell tuning than historically-stable landmarks (Knierim, Kudrimoti, and McNaughton, 1995), but new, environment-specific landmarks have strong roles in spatial representation (Geiller et al., 2017), Cue spatial

stability is not only a memory-dependent effect; in some cases, it seems to be a property of the cue itself. The literature on landmark navigation comparing spatially distal and proximal cues illustrates this concept. The consistent finding is that the position of distal cues, while often smaller than proximal cues and providing information from only the visual modality, correlate more strongly with the activity of have a stronger influence on spatially-modulated cells (place cells and head direction cells) than proximal cues (Zugaro et al., 2001; Yoganarasimha, 2006), with distal cues favored even when performance in a task demands attention to proximal cues (Trullier et al., 1999) or when the distal cues cannot be used for spatial triangulation (Cressant, Muller, and Poucet, 1997). This may be because distal objects are often more reliable for navigation; it takes a greater amount of self-motion and the cue's motion for its direction relative to the observer to change. Because the flexibility of a fully-digital virtual environment theoretically enables high-speed, constant environment changes as part of an experiment, understanding which aspects of an environment change are affected by memory will be helpful for establishing good experimental protocols for spatial navigation studies.

While freely-moving CAVE setups have existed for over twenty-five years, they have to-date occupied a relatively small niche in the virtual reality space, most notably as an implementation for driving and flight simulators, despite the qualities that make them more immersive than other VR solutions (Cruz-Neira et al., 1992; Muhanna, 2015). By designing a single-projector variant, we hope that CAVEs will be reconsidered for future work in animal behavior research, for which a great deal remains to be discovered. Many experimental paradigms are being proposed regularly since the completion of the project: Optic flow manipulations to explore the visual contributions to self-motion perception, operant conditioning in different virtual environments to study environmental effects of context-dependent learning, and virtual predators stimuli as a pain-free alternative to fear conditioning are all potential research projects in the near future. We hope that the open-source and modular implementation of this project will ease the building of similar systems by other groups, and look forward to innovative and exciting virtual reality projects in the coming years.

Bibliography

- Adams, Rick A., Stewart Shipp, and Karl J. Friston (2013). "Predictions not commands: Active inference in the motor system". In: *Brain Structure and Function* 218.3, pp. 611–643. ISSN: 18632653. DOI: 10.1007/s00429-012-0475-5. arXiv: arXiv:1011.1669v3.
- Aghajani, Zahra M et al. (2014). "Impaired spatial selectivity and intact phase precession in two-dimensional virtual reality". In: *Nature Neuroscience* November, pp. 4–6. ISSN: 1097-6256. DOI: 10.1038/nn.3884.
- Aharoni, Daniel Benjamin (2013). "Rats in Virtual Space : The development and implementation of a multimodal virtual reality system for small animals". PhD thesis. University of California, Los Angeles, p. 135.
- Ahrens, Misha B. et al. (2012). "Brain-wide neuronal dynamics during motor adaptation in zebrafish". In: *Nature* 485.7399, pp. 471–477. ISSN: 00280836. DOI: 10.1038/nature11057. arXiv: arXiv:1011.1669v3.
- Allison, Robert S et al. (2001). "Tolerance of Temporal Delay in Virtual Environments". In: *Virtual Reality*. DOI: 10.1109/VR.2001.913793.
- Angelaki, Dora E. and Kathleen E. Cullen (2008). "Vestibular System: The Many Facets of a Multimodal Sense". In: *Annual Review of Neuroscience* 31.1, pp. 125–150. ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.31.060407.125555.
- Arleo, Angelo et al. (2013). "Optic flow stimuli update anterodorsal thalamus head direction neuronal activity in rats." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 33.42, pp. 16790–5. ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.2698-13.2013.
- Aronov, Dmitriy and David W. Tank (2014). "Engagement of Neural Circuits Underlying 2D Spatial Navigation in a Rodent Virtual Reality System". In: *Neuron* 84.2, pp. 442–456. ISSN: 08966273. DOI: 10.1016/j.neuron.2014.08.042.
- Arriaga, Moises and Edward B. Han (2017). "Dedicated Hippocampal Inhibitory Networks for Locomotion and Immobility". In: *The Journal of Neuroscience* 37.38, pp. 9222–9238. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1076-17.2017.
- Ash, April et al. (2011). "Display Lag and Gain Effects on Vection Experienced by Active Observers". In: *Aviation, Space, and Environmental Medicine* 82.8, pp. 763–769. ISSN: 00956562. DOI: 10.3357/ASEM.3026.2011.
- Attinger, Alexander, Bo Wang, and Georg B. Keller (2017). "Visuomotor Coupling Shapes the Functional Development of Mouse Visual Cortex". In: *Cell* 169.7, 1291–1302.e14. ISSN: 00928674. DOI: 10.1016/j.cell.2017.05.023.

- Ayaz, Asli et al. (2013). "Locomotion Controls Spatial Integration in Mouse Visual Cortex". In: *Current Biology* 23.10, pp. 890–894. ISSN: 09609822. DOI: 10.1016/j.cub.2013.04.012.
- Baird, Emily et al. (2013). "A universal strategy for visually guided landing". In: *Proceedings of the National Academy of Sciences* 110.46, pp. 18686–18691. ISSN: 1091-6490. DOI: 10.1073/pnas.1314311110/-/DCSupplemental.www.pnas.org/cgi/doi/10.1073/pnas.1314311110. arXiv: arXiv:1408.1149.
- Barry, Caswell et al. (2006). "The boundary vector cell model of place cell firing and spatial memory". In: *Reviews in the Neurosciences* 17.1, pp. 71–97.
- Bassett, J P and J S Taube (2001). "Neural correlates for angular head velocity in the rat dorsal tegmental nucleus." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 21.15, pp. 5740–5751. ISSN: 1529-2401.
- Bassett, J. P., M. L. Tullman, and J. S. Taube (2007). "Lesions of the Tegmentomammillary Circuit in the Head Direction System Disrupt the Head Direction Signal in the Anterior Thalamus". In: *Journal of Neuroscience* 27.28, pp. 7564–7577. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.0268-07.2007.
- Bassett, Joshua P et al. (2005). "Passive Movements of the Head Do Not Abolish Anticipatory Firing Properties of Head Direction Cells". In: *Journal of neurophysiology* 93.October 2004, pp. 1304–1316. ISSN: 0022-3077. DOI: 10.1152/jn.00490.2004..
- Becker, Wolfgang et al. (2002). "Fusion of vestibular and podokinesthetic information during self-turning towards instructed targets". In: *Experimental Brain Research* 144.4, pp. 458–474. ISSN: 00144819. DOI: 10.1007/s00221-002-1053-5.
- Bergeron, Mathieu, Catherine L. Lortie, and Matthieu J. Guitton (2015). "Use of Virtual Reality Tools for Vestibular Disorders Rehabilitation: A Comprehensive Analysis". In: *Advances in Medicine* 2015, pp. 1–9. ISSN: 2356-6752. DOI: 10.1155/2015/916735.
- Blair, H T and P E Sharp (1995). "Anticipatory head direction signals in anterior thalamus: evidence for a thalamocortical circuit that integrates angular head motion to compute head direction." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 15.9, pp. 6260–6270. ISSN: 0270-6474.
- Blaser, Rachel and Charles Heyser (2015). "Spontaneous object recognition: a promising approach to the comparative study of memory". In: *Frontiers in Behavioral Neuroscience* 9.July, pp. 1–12. ISSN: 1662-5153. DOI: 10.3389/fnbeh.2015.00183.
- Bohil, Corey J., Bradly Alicea, and Frank A. Biocca (2011). "Virtual reality in neuroscience research and therapy". In: *Nature Reviews Neuroscience* 12.12. ISSN: 1471003X. DOI: 10.1038/nrn3122.
- Brainard, David H. (1997). "The Psychophysics Toolbox". In: *Spatial Vision* 10.4, pp. 433–436. ISSN: 0169-1015. DOI: 10.1163/156856897X00357. arXiv: arXiv:1011.1669v3.
- Burn, Charlotte C. (2008). "What is it like to be a rat? Rat sensory perception and its implications for experimental design and rat welfare". In: *Applied Animal Behaviour Science* 112.1-2, pp. 1–32. ISSN: 01681591. DOI: 10.1016/j.applanim.2008.02.007.

- Carandini, Matteo and Anne K Churchland (2013). "Probing perceptual decisions in rodents". In: *Nature Neuroscience* 16.7, pp. 824–831. ISSN: 1097-6256. DOI: 10.1038/nn.3410.
- Carrel, Judith S (1972). "An Improved Treading Device for Tethered Insects". In: *Science* 175.4027, p. 1279. DOI: 10.1126/science.175.4027.1279.
- Carriot, J., J. X. Brooks, and K. E. Cullen (2013). "Multimodal Integration of Self-Motion Cues in the Vestibular System: Active versus Passive Translations". In: *Journal of Neuroscience* 33.50, pp. 19555–19566. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.3051-13.2013.
- Chen, Guifen et al. (2013). "How vision and movement combine in the hippocampal place code." In: *Proceedings of the National Academy of Sciences of the United States of America* 110.1, pp. 378–83. ISSN: 1091-6490. DOI: 10.1073/pnas.1215834110.
- Chen, Guifen et al. (2018). "Spatial cell firing during virtual navigation of open arenas by head-restrained mice". In: *bioRxiv*. DOI: 10.1101/246744.
- Chiappe, M. Eugenia et al. (2010). "Walking modulates speed sensitivity in drosophila motion vision". In: *Current Biology* 20.16, pp. 1470–1475. ISSN: 09609822. DOI: 10.1016/j.cub.2010.06.072.
- Crapse, Trinity B. and Marc A. Sommer (2008). "Corollary discharge across the animal kingdom". In: *Nature Reviews Neuroscience* 9.8, pp. 587–600. ISSN: 1471003X. DOI: 10.1038/nrn2457. arXiv: 15334406.
- Cressant, A, R U Muller, and B Poucet (1997). "Failure of centrally placed objects to control the firing fields of hippocampal place cells." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 17.7, pp. 2531–2542. ISSN: 0270-6474.
- Cruz-Neira, Carolina et al. (1992). "The CAVE: audio visual experience automatic virtual environment". In: *Communications of the ACM* 35.6, pp. 64–72. ISSN: 00010782. DOI: 10.1145/129888.129892.
- Cullen, Kathleen E. and Jefferson E Roy (2004). "Signal Processing in the Vestibular System During Active Versus Passive Head Movements". In: *Journal of Neurophysiology* 91.5, pp. 1919–1933. ISSN: 0022-3077. DOI: 10.1152/jn.00988.2003.
- Cullen, Kathleen E. and Jeffrey S. Taube (2017). "Our sense of direction: Progress, controversies and challenges". In: *Nature Neuroscience* 20.11, pp. 1465–1473. ISSN: 15461726. DOI: 10.1038/nn.4658.
- Cushman, Jesse D. et al. (2013). "Multisensory Control of Multimodal Behavior: Do the Legs Know What the Tongue Is Doing?" In: *PLoS ONE* 8.11. Ed. by Anna Dunaevsky, e80465. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0080465.
- Del Grosso, Nicholas A et al. (2017). "Virtual Reality system for freely-moving rodents." In: *bioRxiv*. DOI: <https://doi.org/10.1101/161232>.
- Di Luca, Massimiliano (2010). "New Method to Measure End-to-End Delay of Virtual Reality". In: *Presence: Teleoperators and Virtual Environments* 19.6, pp. 569–584. ISSN: 1054-7460. DOI: 10.1162/pres_a_00023.

- Dittmar, L. et al. (2010). "Goal seeking in honeybees: matching of optic flow snapshots?" In: *Journal of Experimental Biology* 213.17, pp. 2913–2923. ISSN: 0022-0949. DOI: 10.1242/jeb.043737.
- Dombeck, Daniel A. and Michael B. Reiser (2012). "Real neuroscience in virtual worlds". In: *Current Opinion in Neurobiology* 22.1, pp. 3–10. ISSN: 09594388. DOI: 10.1016/j.conb.2011.10.015.
- Dombeck, Daniel a et al. (2010). "Functional imaging of hippocampal place cells at cellular resolution during virtual navigation." In: *Nature neuroscience* 13.11, pp. 1433–40. ISSN: 1546-1726. DOI: 10.1038/nn.2648.
- Domnisoru, Cristina, Amina A Kinkhabwala, and David W Tank (2013). "Membrane potential dynamics of grid cells". In: *Nature* 495.7440, pp. 199–204. ISSN: 0028-0836. DOI: 10.1038/nature11973. arXiv: NIHMS150003.
- Douglas, R. M. et al. (2005). "Independent visual threshold measurements in the two eyes of freely moving rats and mice using a virtual-reality optokinetic system". In: *Visual Neuroscience* 22.5, pp. 677–684. ISSN: 09525238. DOI: 10.1017/S0952523805225166.
- Draper, Mark Heider (1998). "The Adaptive Effects Of Virtual Interfaces: Vestibulo-Ocular Reflex and Simulator Sickness by". PhD thesis. University of Washington.
- Driscoll, Laura N. et al. (2017). "Dynamic Reorganization of Neuronal Activity Patterns in Parietal Cortex". In: *Cell* 170.5, 986–999.e16. ISSN: 10974172. DOI: 10.1016/j.cell.2017.07.021.
- Engel, Andreas K. et al. (2013). "Where's the action? The pragmatic turn in cognitive science". In: *Trends in Cognitive Sciences* 17.5, pp. 202–209. ISSN: 13646613. DOI: 10.1016/j.tics.2013.03.006.
- Feldman, A. G. (2015). *Referent control of action and perception: Challenging conventional theories in behavioral neuroscience*. Springer. ISBN: 978-1-4939-2736-4.
- Fenk, L. M., T. Hoinkes, and A. Schmid (2010). "Vision as a third sensory modality to elicit attack behavior in a nocturnal spider". In: *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology* 196.12, pp. 957–961. ISSN: 03407594. DOI: 10.1007/s00359-010-0575-8.
- Foster, Tom, Carl Castro, and McNaughton Bruce (1989). "Spatial Selectivity of Rat Hippocampal Neurons: Dependence on Preparedness of Movement". In: *Science Reports* 244, pp. 1580–1581.
- Frissen, Ilja et al. (2011). "Integration of vestibular and proprioceptive signals for spatial updating". In: *Experimental Brain Research* 212.2, pp. 163–176. ISSN: 00144819. DOI: 10.1007/s00221-011-2717-9.
- Fritz, N.E., F.M. Cheek, and D.S. Nichols-Larsen (2015). "Motor-Cognitive Dual-Task Training in Neurologic Disorders: A Systematic Review". In: *J Neurol Phys Ther.* 39.3, pp. 142–153. DOI: 10.1097/NPT.000000000000090.Motor-Cognitive.
- Fry, S. N. et al. (2004). "Context-dependent stimulus presentation to freely moving animals in 3D". In: *Journal of Neuroscience Methods* 135.1-2, pp. 149–157. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2003.12.012.

- Fry, S. N. et al. (2009). "Visual control of flight speed in *Drosophila melanogaster*". In: *Journal of Experimental Biology* 212.8, pp. 1120–1130. ISSN: 0022-0949. DOI: 10.1242/jeb.020768.
- Fry, Steven N. et al. (2008). "TrackFly: Virtual reality for a behavioral system analysis in free-flying fruit flies". In: *Journal of Neuroscience Methods* 171.1, pp. 110–117. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2008.02.016.
- Furtak, Sharon C et al. (2009). "The Floor Projection Maze: A novel behavioral apparatus for presenting visual stimuli to rats." In: *Journal of neuroscience methods* 181.1, pp. 82–8. ISSN: 1872-678X. DOI: 10.1016/j.jneumeth.2009.04.023.
- Gegenfurtner, Karl R. (2016). "The Interaction Between Vision and Eye Movements†". In: *Perception* 45.12, pp. 1333–1357. ISSN: 03010066. DOI: 10.1177/0301006616657097.
- Geiller, Tristan et al. (2017). "Place cells are more strongly tied to landmarks in deep than in superficial CA1". In: *Nature Communications* 8. ISSN: 2041-1723. DOI: 10.1038/ncomms14531.
- Geva-Sagiv, Maya et al. (2016). "Hippocampal global remapping for different sensory modalities in flying bats". In: *Nature Neuroscience* 19.7, pp. 952–958. ISSN: 15461726. DOI: 10.1038/nn.4310.
- Gianelli, Sam, Bruce Harland, and Jean Marc Fellous (2018). "A new rat-compatible robotic framework for spatial navigation behavioral experiments". In: *Journal of Neuroscience Methods* 294, pp. 40–50. ISSN: 1872678X. DOI: 10.1016/j.jneumeth.2017.10.021.
- Gibson, Brett, William N. Butler, and Jeffery S. Taube (2013). "The head-direction signal is critical for navigation requiring a cognitive map but not for learning a spatial habit". In: *Current Biology* 23.16, pp. 1536–1540. ISSN: 09609822. DOI: 10.1016/j.cub.2013.06.030. arXiv: NIHMS150003.
- Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. Boston: Houghton Mifflin. ISBN: 978-0313239618.
- Goodridge, J P et al. (1998). "Cue control and head direction cells." In: *Behavioral neuroscience* 112.4, pp. 749–761. ISSN: 0735-7044. DOI: 10.1037/0735-7044.112.4.749.
- Gray, John R, Vincent Pawlowski, and Mark A Willis (2002). "A method for recording behavior and multineuronal CNS activity from tethered insects flying in virtual space". In: *Journal of neuroscience methods* 120, pp. 211–223.
- Guo, Zengcai V et al. (2014). "Procedures for behavioral experiments in head-fixed mice." In: *PloS one* 9.2, e88678. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0088678.
- Hainmueller, Thomas and Marlene Bartos (2018). "Parallel emergence of stable and dynamic memory engrams in the hippocampus". In: *Nature* 558.7709, pp. 292–296. ISSN: 0028-0836. DOI: 10.1038/s41586-018-0191-2.
- Harvey, Christopher D, Philip Coen, and David W Tank (2012). "Choice-specific sequences in parietal cortex during a virtual-navigation decision task." In: *Nature* 484.7392, pp. 62–8. ISSN: 1476-4687. DOI: 10.1038/nature10918.

- Harvey, Christopher D et al. (2009). "Intracellular dynamics of hippocampal place cells during virtual navigation." In: *Nature* 461.7266, pp. 941–6. ISSN: 1476-4687. DOI: 10.1038/nature08499.
- Heys, James G., Krsna V. Rangarajan, and Daniel A. Dombeck (2014). "The Functional Micro-organization of Grid Cells Revealed by Cellular-Resolution Imaging". In: *Neuron* 84.5, pp. 1079–1090. ISSN: 08966273. DOI: 10.1016/j.neuron.2014.10.048.
- Hölscher, C et al. (2005). "Rats are able to navigate in virtual environments." In: *The Journal of experimental biology* 208, pp. 561–9. ISSN: 0022-0949. DOI: 10.1242/jeb.01371.
- Hu, H.H. et al. (2000). "Visual cues for imminent object contact in realistic virtual environments". In: *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)* November 2000, pp. 179–185, DOI: 10.1109/VISUAL.2000.885692.
- Huang, Fu-Chung, David Luebke, and Gordon Wetzstein (2015). "The light field stereoscope". In: *ACM SIGGRAPH 2015 Emerging Technologies on - SIGGRAPH '15*. New York, New York, USA: ACM Press, pp. 1–1. ISBN: 9781450336352. DOI: 10.1145/2782782.2792493.
- Hubona, Geoffrey S et al. (1999). "The relative contributions of stereo, lighting, and background scenes in promoting 3D depth visualization". In: *ACM Transactions on Computer-Human Interaction* 6.3, pp. 214–242. ISSN: 10730516. DOI: 10.1145/329693.329695.
- Jacobs, Joshua et al. (2013). "Direct recordings of grid-like neuronal activity in human spatial navigation." In: *Nature neuroscience* 16.9, pp. 1188–90. ISSN: 1546-1726. DOI: 10.1038/nn.3466.
- Jangraw, David C. et al. (2014). "NEDE: An open-source scripting suite for developing experiments in 3D virtual environments". In: *Journal of Neuroscience Methods* 235, pp. 245–251. ISSN: 1872678X. DOI: 10.1016/j.jneumeth.2014.06.033.
- Jeffery, Kathryn J. and John M. O'Keefe (1999). "Learned interaction of visual and idiothetic cues in the control of place field orientation". In: *Experimental Brain Research* 127.2, pp. 151–161. ISSN: 00144819. DOI: 10.1007/s002210050785.
- Jerald, Jason (2004). "Latency Compensation for Head-Mounted Virtual Reality". In: Jezeq, Karel et al. (2011). "Theta-paced flickering between place-cell maps in the hippocampus." In: *Nature* 478.7368, pp. 246–9. ISSN: 1476-4687. DOI: 10.1038/nature10439.
- Jouary, Adrien et al. (2016). "A 2D virtual reality system for visual goal-driven navigation in zebrafish larvae". In: *Scientific Reports* 6.1, p. 34015. ISSN: 2045-2322. DOI: 10.1038/srep34015.
- Jung, S. N., A. Borst, and J. Haag (2011). "Flight Activity Alters Velocity Tuning of Fly Motion-Sensitive Neurons". In: *Journal of Neuroscience* 31.25, pp. 9231–9237. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1138-11.2011.

- Jürgens, Reinhart and Wolfgang Becker (2006). "Perception of angular displacement without landmarks: Evidence for Bayesian fusion of vestibular, optokinetic, podokines-
thetic, and cognitive information". In: *Experimental Brain Research* 174.3, pp. 528–
543. ISSN: 00144819. DOI: 10.1007/s00221-006-0486-7.
- Kaupert, Ursula et al. (2017). "Spatial cognition in a virtual reality home-cage exten-
sion for freely moving rodents". In: *Journal of Neurophysiology* 117.4, pp. 1736–1748.
ISSN: 0022-3077. DOI: 10.1152/jn.00630.2016.
- Kay, Kenneth et al. (2016). "A hippocampal network for spatial coding during immo-
bility and sleep". In: *Nature* 531.7593, pp. 185–190. ISSN: 14764687. DOI: 10.1038/
nature17144. arXiv: 15334406.
- Keller, Georg B, Tobias Bonhoeffer, and Mark Hübener (2012). "Sensorimotor mis-
match signals in primary visual cortex of the behaving mouse." In: *Neuron* 74.5,
pp. 809–15. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2012.03.040.
- Kerfoot, William B (1968). "Orientometer for Study of Insect Behavior". In: *Science*
162.3852, pp. 477–477. ISSN: 0036-8075. DOI: 10.1126/science.162.3852.477.
- Kijima, Ryugo, Yamada Eijiroh, and Takeo Ojika (2001). "A Development of Reflex
HMD: HMD with time delay compensation capability". In: *Proceedings of Interna-
tional Symposium on Mixed Reality* 2001, pp. 14–15.
- Kim, Juno et al. (2015). "The Oculus Rift: A cost-effective tool for studying visual-
vestibular interactions in self-motion perception". In: *Frontiers in Psychology* 6.MAR,
pp. 1–7. ISSN: 16641078. DOI: 10.3389/fpsyg.2015.00248.
- Kleinfeld, David et al. (2014). "More than a rhythm of life: breathing as a binder
of orofacial sensation". In: *Nature Neuroscience* 17.5, pp. 647–651. ISSN: 1097-6256.
DOI: 10.1038/nn.3693.
- Knierim, J J, H S Kudrimoti, and B L McNaughton (1995). "Place cells, head direction
cells, and the learning of landmark stability." In: *The Journal of Neuroscience* 15.3,
pp. 1648–1659. ISSN: 0270-6474.
- (1998). "Interactions between idiothetic cues and external landmarks in the control
of place cells and head direction cells." In: *Journal of neurophysiology* 80.1, pp. 425–
446. ISSN: 0022-3077.
- Knierim, James J. and Derek A. Hamilton (2011). "Framing Spatial Cognition: Neural
Representations of Proximal and Distal Frames of Reference and Their Roles in
Navigation". In: *Physiological Reviews* 91.4, pp. 1245–1279. ISSN: 0031-9333. DOI:
10.1152/physrev.00021.2010. arXiv: NIHMS150003.
- Kötter, Rolf (2009). "A primer of visual stimulus presentation software". In: *Frontiers
in Neuroscience* 3.2, pp. 163–164. ISSN: 16624548. DOI: 10.3389/neuro.01.021.
2009.
- Kramida, Gregory and Amitabh Varshney (2015). "Resolving the Vergence-Accommodation
Conflict in Head Mounted Displays". In: *IEEE Transactions on Visualization and
Computer Graphics* 22.7, pp. 1–16. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.
2473855. arXiv: arXiv:1011.1669v3.

- Krause, Florian and Oliver Lindemann (2014). "Experiment: A Python library for cognitive and neuroscientific experiments". In: *Behavior Research Methods* 46.2, pp. 416–428. ISSN: 1554-3528. DOI: 10.3758/s13428-013-0390-6.
- Lasztóczy, Bálint and Thomas Klausberger (2014). "Layer-Specific GABAergic Control of Distinct Gamma Oscillations in the CA1 Hippocampus." In: *Neuron* 81.5, pp. 1126–39. ISSN: 1097-4199. DOI: 10.1016/j.neuron.2014.01.021.
- Lee, Hsiao-Yu et al. (2007). "Development of Virtual Reality Environment for Tracking Rat Behavior". In: *Journal of Medical and Biological Engineering* 2, pp. 73–78.
- Leinweber, Marcus et al. (2017). "A Sensorimotor Circuit in Mouse Cortex for Visual Flow Predictions". In: *Neuron* 95.6, 1420–1432.e5. ISSN: 10974199. DOI: 10.1016/j.neuron.2017.08.036.
- Markus, E. J. et al. (1994). "Spatial information content and reliability of hippocampal CA1 neurons: Effects of visual input". In: *Hippocampus* 4.4, pp. 410–421. ISSN: 10509631. DOI: 10.1002/hipo.450040404.
- McCrea, R a et al. (1999). "Firing behavior of vestibular neurons during active and passive head movements: vestibulo-spinal and other non-eye-movement related neurons." In: *Journal of neurophysiology* 82.1, pp. 416–428. ISSN: 0022-3077.
- Meer, M. A. A. van der et al. (2007). "Anticipation in the Rodent Head Direction System Can Be Explained by an Interaction of Head Movements and Vestibular Firing Properties". In: *Journal of Neurophysiology* 98.4, pp. 1883–1897. ISSN: 0022-3077. DOI: 10.1152/jn.00233.2007.
- Minderer, Matthias et al. (2016). "Virtual reality explored". In: *Nature* 533.7603, pp. 324–325. ISSN: 0028-0836. DOI: 10.1038/nature17899.
- Morillon, Benjamin et al. (2015). "Predictive motor control of sensory dynamics in auditory active sensing". In: *Current Opinion in Neurobiology* 31, pp. 230–238. ISSN: 18736882. DOI: 10.1016/j.conb.2014.12.005. arXiv: 15334406.
- Muhanna, Muhanna A. (2015). "Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions". In: *Journal of King Saud University - Computer and Information Sciences* 27.3, pp. 344–361. ISSN: 22131248. DOI: 10.1016/j.jksuci.2014.03.023.
- Muir, Gary M et al. (2009). "Disruption of the head direction cell signal after occlusion of the semicircular canals in the freely moving chinchilla." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 29.46, pp. 14521–33. ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.3450-09.2009.
- Nakayasu, Tomohiro et al. (2017). "Three-dimensional computer graphic animations for studying social approach behaviour in medaka fish: Effects of systematic manipulation of morphological and motion cues". In: *PLoS ONE* 12.4, pp. 1–17. ISSN: 19326203. DOI: 10.1371/journal.pone.0175059.
- Niell, Cristopher M. and Michael P. Stryker (2010). "Modulation of Visual Responses by Behavioral State in Mouse Visual Cortex". In: *Neuron* 65.4, pp. 472–479. ISSN: 08966273. DOI: 10.1016/j.neuron.2010.01.033. arXiv: arXiv:1011.1669v3.

- Nityananda, Vivek et al. (2016). "Insect stereopsis demonstrated using a 3D insect cinema". In: *Scientific Reports* 6.1, p. 18718. ISSN: 2045-2322. DOI: 10.1038/srep18718.
- Normand, Jean-Marie et al. (2012). "Beaming into the rat world: enabling real-time interaction between rat and human each at their own scale." In: *PloS one* 7.10, e48331. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0048331.
- Orger, Michael B. et al. (2008). "Control of visually guided behavior by distinct populations of spinal projection neurons". In: *Nature Neuroscience* 11.3, pp. 327–333. ISSN: 10976256. DOI: 10.1038/nn2048.
- Peirce, Jonathan W. (2007). "PsychoPy—Psychophysics software in Python". In: *Journal of Neuroscience Methods* 162.1-2, pp. 8–13. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2006.11.017.
- Phong, Bui Tuong (1975). "Illumination for computer generated pictures". In: *Communications of the ACM* 18.6, pp. 311–317. ISSN: 00010782. DOI: 10.1145/360825.360839.
- Pinto, Lucas et al. (2018). "An Accumulation-of-Evidence Task Using Visual Pulses for Mice Navigating in Virtual Reality". In: *Frontiers in Behavioral Neuroscience* 12, pp. 1–19. ISSN: 1662-5153. DOI: 10.3389/fnbeh.2018.00036.
- Posani, Lorenzo et al. (2017). "Functional connectivity models for decoding of spatial representations from hippocampal CA1 recordings". In: *Journal of Computational Neuroscience* 43.1, pp. 17–33. ISSN: 15736873. DOI: 10.1007/s10827-017-0645-9.
- Radvansky, Brad A. and Daniel A. Dombeck (2018). "An olfactory virtual reality system for mice". In: *Nature Communications* 9.1, p. 839. ISSN: 2041-1723. DOI: 10.1038/s41467-018-03262-4.
- Ravassard, Pascal et al. (2013). "Multisensory control of hippocampal spatiotemporal selectivity." In: *Science* 340.6138, pp. 1342–6. ISSN: 1095-9203. DOI: 10.1126/science.1232655.
- Reiser, Michael B. and Michael H. Dickinson (2008). "A modular display system for insect behavioral neuroscience". In: *Journal of Neuroscience Methods* 167.2, pp. 127–139. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2007.07.019.
- Roy, J E and K E Cullen (2001). "Selective processing of vestibular reafference during self-generated head motion." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 21.6, pp. 2131–2142. ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.2131-01.2001.
- Sato, Masaaki et al. (2017). "Hippocampus-Dependent Goal Localization by Head-Fixed Mice in Virtual Reality". In: *eneuro* 4.3, ENEURO.0369–16.2017. ISSN: 2373-2822. DOI: 10.1523/ENEURO.0369-16.2017.
- Save, E et al. (1998). "Spatial firing of hippocampal place cells in blind rats." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 18.June 2015, pp. 1818–1826. ISSN: 0270-6474.
- Scarfe, Peter and Andrew Glennerster (2015). "Using high-fidelity virtual reality to study perception in freely moving observers". In: *Journal of Vision* 15.9, p. 3. ISSN: 1534-7362. DOI: 10.1167/15.9.3.

- Schaefer, Andreas T. and Adam Claridge-Chang (2012). "The surveillance state of behavioral automation". In: *Current Opinion in Neurobiology* 22.1, pp. 170–176. ISSN: 09594388. DOI: 10.1016/j.conb.2011.11.004.
- Schlick, Christophe (1994). "A Survey of Shading and Reflectance Models". In: *Computer Graphics Forum* 13.2, pp. 121–131. ISSN: 0167-7055. DOI: 10.1111/1467-8659.1320121.
- Schmidt-Hieber, Christoph and Michael Häusser (2013). "Cellular mechanisms of spatial navigation in the medial entorhinal cortex". In: *Nature Neuroscience* 16.3, pp. 325–331. ISSN: 10976256. DOI: 10.1038/nn.3340.
- Schrater, Paul R. (2000). "How Optimal Depth Cue Integration Depends on the Task". In: *International Journal of Computer Vision* 40.1, pp. 71–89. ISSN: 09205691. DOI: 10.1023/A:1026557704054.
- Schulze, Jürgen P. et al. (2013). "CalVR: an advanced open source virtual reality software framework". In: *Proc. SPIE*. Ed. by Margaret Dolinsky and Ian E. McDowall. Vol. 8649, p. 864902. ISBN: 9780819494221. DOI: 10.1117/12.2005241.
- Schwarz, Cornelius et al. (2010). "The head-fixed behaving rat—procedures and pitfalls." In: *Somatosensory & motor research* 27.4, pp. 131–48. ISSN: 1369-1651. DOI: 10.3109/08990220.2010.513111.
- Seeber, Bernhard U., Stefan Kerber, and Ervin R. Hafter (2010). "A system to simulate and reproduce audio–visual environments for spatial hearing research". In: *Hearing Research* 260.1-2, pp. 1–10. ISSN: 03785955. DOI: 10.1016/j.heares.2009.11.004.
- Sheffield, Mark E J and Daniel A Dombek (2015a). "The binding solution?" In: *Nature Neuroscience* 18.8, pp. 1060–1062. ISSN: 1097-6256. DOI: 10.1038/nn.4075.
- Sheffield, Mark E.J. and Daniel A. Dombek (2015b). "Calcium transient prevalence across the dendritic arbour predicts place field properties". In: *Nature* 517.7533, pp. 200–204. ISSN: 14764687. DOI: 10.1038/nature13871.
- Silva, Alcino J. (2017). "Miniaturized two-photon microscope: Seeing clearer and deeper into the brain". In: *Light: Science and Applications* 6.8. ISSN: 20477538. DOI: 10.1038/lsa.2017.104.
- Smear, Matthew et al. (2013). "Multiple perceptible signals from a single olfactory glomerulus". In: *Nature Neuroscience* 16.11, pp. 1687–1691. ISSN: 10976256. DOI: 10.1038/nn.3519.
- Sofroniew, Nicholas J et al. (2014). "Natural Whisker-Guided Behavior by Head-Fixed Mice in Tactile Virtual Reality". In: *The Journal of Neuroscience* 34.29, pp. 9537–9550. DOI: 10.1523/JNEUROSCI.0712-14.2014.
- Sofroniew, Nicholas James et al. (2015). "Neural coding in barrel cortex during whisker-guided locomotion". In: *eLife* 4.DECEMBER2015, pp. 1–19. ISSN: 2050084X. DOI: 10.7554/eLife.12559. arXiv: arXiv:1011.1669v3.
- Solway, Alec, Jonathan F. Miller, and Michael J. Kahana (2013). "PandaEPL: A library for programming spatial navigation experiments". In: *Behavior Research Methods* 45.4, pp. 1293–1312. ISSN: 1554-3528. DOI: 10.3758/s13428-013-0322-5.

- Stackman, R. W. and J. S. Taube (1997). "Firing properties of head direction cells in the rat anterior thalamic nucleus: dependence on vestibular input." In: *The Journal of Neuroscience* 17.11, pp. 4349–4358. ISSN: 0270-6474.
- Stackman, Robert W, Ann S Clark, and Jeffrey S Taube (2002). "Hippocampal spatial representations require vestibular input." In: *Hippocampus* 12.3, pp. 291–303. ISSN: 1050-9631. DOI: 10.1002/hipo.1112.
- Stackman, Robert W et al. (2003). "Passive transport disrupts directional path integration by rat head direction cells." In: *Journal of neurophysiology* 90.5, pp. 2862–2874. ISSN: 0022-3077. DOI: 10.1152/jn.00346.2003.
- Stowers, John R. et al. (2014). "Reverse engineering animal vision with virtual reality and genetics". In: *Computer* 47.7, pp. 38–45. ISSN: 00189162. DOI: 10.1109/MC.2014.190.
- Stowers, John R. et al. (2017). "Virtual reality for freely moving animals". In: *Nature Methods* 14.10, pp. 995–1002. ISSN: 15487105. DOI: 10.1038/nmeth.4399.
- Strauss, Roland, Stefan Schuster, and Karl Georg Götz (1997). "Processing of artificial visual feedback in the walking fruit fly drosophila melanogaster". In: *The Journal of experimental biology* 200, pp. 1281–1296.
- Straw, Andrew D (2008). "Vision Egg: an open-source library for realtime visual stimulus generation". In: *Frontiers in neuroinformatics* 2.November, pp. 1–10. DOI: 10.3389/neuro.11.004.2008.
- Straw, Andrew D., Serin Lee, and Michael H. Dickinson (2010). "Visual control of altitude in flying drosophila". In: *Current Biology* 20.17, pp. 1550–1556. ISSN: 09609822. DOI: 10.1016/j.cub.2010.07.025.
- Stringer, Carsen et al. (2018). "Spontaneous behaviors drive multidimensional, brain-wide neural activity". In: *bioRxiv*, pp. 1–26. DOI: 10.1101/306019.
- Tarr, M J and W H Warren (2002). "Virtual reality in behavioral neuroscience and beyond". In: *Nat Neurosci* 5 Suppl, pp. 1089–92. ISSN: 1097-6256 (Print) 1097-6256 (Linking). DOI: 10.1038/nn948.
- Taube, J S and H L Burton (1995). "Head direction cell activity monitored in a novel environment and during a cue conflict situation." In: *Journal of Neurophysiology* 74.5, pp. 1953–1971. ISSN: 0022-3077.
- Taube, Jeffrey S. (2007). "The Head Direction Signal: Origins and Sensory-Motor Integration". In: *Annual Review of Neuroscience* 30.1, pp. 181–207. ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.29.051605.112854.
- Telford, Laura, Ian P. Howard, and Masao Ohmi (1995). "Heading judgments during active and passive self-motion". In: *Experimental Brain Research* 104.3, pp. 502–510. ISSN: 00144819. DOI: 10.1007/BF00231984.
- Tennant, Sarah A. et al. (2018). "Stellate Cells in the Medial Entorhinal Cortex Are Required for Spatial Learning". In: *Cell Reports* 22.5, pp. 1313–1324. ISSN: 22111247. DOI: 10.1016/j.celrep.2018.01.005.
- Thurley, Kay and Aslı Ayaz (2016). "Virtual Reality Systems for Rodents". In: *Current Zoology* 0.0, pp. 1–11. DOI: <http://dx.doi.org/10.1093/cz/zow070>.

- Thurley, Kay et al. (2014). "Mongolian gerbils learn to navigate in complex virtual spaces." In: *Behavioural brain research* 266, pp. 161–8. ISSN: 1872-7549. DOI: 10.1016/j.bbr.2014.03.007.
- Trullier, O. et al. (1999). "Hippocampal neuronal position selectivity remains fixed to room cues only in rats alternating between place navigation and beacon approach tasks". In: *European Journal of Neuroscience* 11.12, pp. 4381–4388. ISSN: 0953816X. DOI: 10.1046/j.1460-9568.1999.00839.x.
- Villette, Vincent et al. (2015). "Internally Recurring Hippocampal Sequences as a Population Template of Spatiotemporal Information". In: *Neuron* 88.2, pp. 357–366. ISSN: 10974199. DOI: 10.1016/j.neuron.2015.09.052.
- Walk, Richard D. and Eleanor J. Gibson (1961). "A comparative and analytical study of visual depth perception." In: *Psychological Monographs: General and Applied* 75.15, pp. 1–44. ISSN: 0096-9753. DOI: 10.1037/h0093827.
- Warren, William H and Daniel J Hannon (1988). "Direction of self-motion is perceived from optic flow". In: *Nature* 336, pp. 403–405.
- Westerman, S.J. and T. Cribbin (1998). "Individual differences in the use of depth cues: implications for computer- and video-based tasks". In: *Acta Psychologica* 99.3, pp. 293–310. ISSN: 00016918. DOI: 10.1016/S0001-6918(98)00016-X.
- Wiener, Sidney I., Alain Berthoz, and Michaël B. Zugaro (2002). "Multisensory processing in the elaboration of place and head direction responses by limbic system neurons". In: *Cognitive Brain Research* 14, pp. 75–90. ISSN: 09266410. DOI: 10.1016/S0926-6410(02)00062-9.
- Yoder, Ryan M and Jeffrey S Taube (2009). "Head direction cell activity in mice: robust directional signal depends on intact otolith organs". In: *Journal of Neurosci* 29.4, pp. 1061–1076. DOI: 10.1523/JNEUROSCI.1679-08.2009.Head.
- Yoganarasimha, D (2006). "Head Direction Cell Representations Maintain Internal Coherence during Conflicting Proximal and Distal Cue Rotations: Comparison with Hippocampal Place Cells". In: *Journal of Neuroscience* 26.2, pp. 622–631. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.3885-05.2006.
- Young, B J, G D Fox, and H Eichenbaum (1994). "Correlates of hippocampal complex-spike cell activity in rats performing a nonspatial radial maze task." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 14.November, pp. 6553–6563. ISSN: 0270-6474. DOI: 10.1016/S0896-6273(00)80909-5.
- Zheng, Feng et al. (2014). "Minimizing latency for augmented reality displays: Frames considered harmful". In: *ISMAR 2014 - IEEE International Symposium on Mixed and Augmented Reality - Science and Technology 2014, Proceedings* 2.1, pp. 195–200. DOI: 10.1109/ISMAR.2014.6948427.
- Zoccolan, Davide (2015). "Invariant visual object recognition and shape processing in rats". In: *Behavioural Brain Research* 285, pp. 10–33. ISSN: 18727549. DOI: 10.1016/j.bbr.2014.12.053.

Zugaro, M B et al. (2001). "Active locomotion increases peak firing rates of anterodorsal thalamic head direction cells." In: *Journal of neurophysiology* 86.2, pp. 692–702. ISSN: 0022-3077.

LIST OF PUBLICATIONS

JOURNAL PUBLICATIONS

Nicholas A. Del Grosso, Justin J. Graboski, Weiwei Chen, Eduardo Blanco Hernández, Anton Sirota. "Virtual Reality system for freely-moving rodents." bioRxiv 161232. July 2017; doi=<https://doi.org/10.1101/161232>

Broetz D., Del Grosso, N.A., Rea M., Ramos-Murguialday, A., Soekadar S.R., Birbaumer, N. "A New Hand Assessment Instrument for Severely Affected Stroke Patients." *Journal of Neurorehabilitation*. 2014; 34(3), 409-27.

Benoit, J.B., Del Grosso, N.A., Yoder, J.A., Denlinger, D.L. "Resistance to Dehydration between Bouts of Blood Feeding in the Bed Bug, *Cimex Lectularius*, is Enhanced by Water Conservation, Aggregation, and Quiescence." *American Journal of Tropical Medical Hygiene*. May 2007; 76(5), 987-93.

August 5, 2018

CURRICULUM VITAE

EDUCATION

Oct 2014 - Present	PhD. Candidate	Graduate School of Systemic Neurosciences, Lüdwig-Maximilians Universität
Aug 2012	M.Sc. Neuroscience	Graduate Training Centre of Neuroscience, Eberhard Karls Universität Tübingen
May 2010	B.Sc. Psychology	Wittenberg University

RESEARCH EXPERIENCE

May 2013 - Present	Ludwig-Maximilians Universität	Prof. Dr. Anton Sirota Programmed a 3D graphics engine in Python to build virtual reality system for freely moving rats, supervised students in programming, engineering, and cognitive science projects, organized weekly journal clubs, and ordered new equipment, trained rodents to perform behavioral tasks, and performed surgery on said rodents as part of brain research.
Aug 2012 - May 2013	Universität Tübingen	Prof. Dr. Christoph Braun Wrote a research grant to study the top-down and bottom-up interactions by computational modeling information propagation in early sensory pathways as measured by MEG, designed and administrated an institute wiki, organized a student lecture series, and supervised two students' EEG research projects.
Nov 2011 - July 2012	Universität Tübingen	Prof. Dr. Niels Birbaumer Programmed in Matlab a time-frequency and evoked potential analysis on three years' worth of MEG data assessing longitudinal changes in stroke patients receiving physiotherapy.
Oct 2012 - Nov 2012	Universität Tübingen	Prof. Dr. Cornelius Schwarz Trained rats to perform whisking in response to barrel cortex stimulation viachronically-implanted electrodes, mapping stimulation sensitivity to each cortical layer.
Nov 2010 - March 2011	Universität Tübingen	Dr. Michael Barnett Cowan Programmed an online EMG classifier in Matlab and Simulink to accurately detect finger movements within milliseconds for EEG coherence brain-computer interface training.
Dec. 2009 - Aug. 2010	Wittenberg University	Prof. Dr. Josephine Wilson Built an NI-DAQ EEG system, programmed online analysis and data acquisition in Matlab and LabView, and confirmed its functionality in three different experiments. As a senior lab assistant, also worked as an aid for rat neurosurgery and noninvasive electrophysiology (skin conductance, EMG, EKG, and EEG) laboratory course sessions, which included planning and giving demonstrations on each method above.
June-Aug 2008 - June 2009	Duke University	Prof. Dr. Jennifer Groh Trained Macaque monkeys to perform visual saccade tasks while mapping receptive fields in superior and inferior colliculus.
Aug 2007 - Dec. 2009	Wittenberg University	Prof. Dr. Michael Anes Conducted three behavioral psychophysics studies on the hemispheric lateralization of face perception. Tasks included programming stimulus sequences in SuperLab, patient recruitment and management, data collection, and conference poster preparation.
Nov 2006 - March 2007	Wittenberg University	Prof. Dr. Jay Yoder Measured dessication rates in the bed bug and isolated fungal growth in three species of cockroach. These studies resulted in a publication in a peer-reviewed journal and a poster presentation at an undergraduate research conference.

Eidesstattliche Versicherung / Affidavit

Hiermit versichere ich an Eides statt, dass ich die vorliegende Dissertation "Design of Virtual Reality Systems for Animal Behavior Research" selbstständig angefertigt habe, mich außer der angegebenen keiner weiteren Hilfsmittel bedient und alle Erkenntnisse, die aus dem Schrifttum ganz oder annähernd übernommen sind, als solche kenntlich gemacht und nach ihrer Herkunft unter Bezeichnung der Fundstelle einzeln nachgewiesen habe.

I hereby confirm that the dissertation "Design of Virtual Reality Systems for Animal Behavior Research" is the result of my own work and that I have only used sources or materials listed and specified in the dissertation.

Nicholas A. Del Grosso

München, den 20 Juli, 2018
Munich, date July 20, 2018

Declaration of Author Contributions

Chapters: Thesis Introduction and Discussion. All work (writing the paper, figure production, and the VR setup component design process) was performed by N.A.D.G.

Chapter: A 3D Graphics Python Package for Cognitive Psychology Experiments.

Authors: Nicholas A. Del Grosso and Anton Sirota

N.A.D.G. designed and implemented the Ratcave graphics software, wrote the paper, and produced the figures and videos. A.S. discussed the software's development and commented on the manuscript.

Chapter: A Virtual Reality System for Freely-Moving Rodents. *Authors:* Nicholas

A. Del Grosso, Justin J. Graboski, Weiwei Chen, Eduardo Blanco-Hernandez, and Anton Sirota

N.A.D.G. designed and implemented the ratCAVE system, designed and performed behavioral experiments in the VR; J.J.G. and E.B.H. performed electrophysiological experiments recordings in the VR and behavioral experiments with real objects exploration; all authors contributed to data analysis; N.A.D.G., E.B.H and A.S. wrote the paper. All authors discussed the results and commented on the manuscript.

First Author: Nicholas A. Del Grosso

Supervisor: Prof. Dr. Anton Sirota

München, den 20 Juli, 2018
Munich, date July 20, 2018

München, den 20 Juli, 2018
Munich, date July 20, 2018