Journal of Neuroscience Methods 135 (2004) 149–157

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Context-dependent stimulus presentation to freely moving animals in 3D

S.N. Fry[a,*], P. Müller[b], H.-J. Baumann[b], A.D. Straw[c], M. Bichsel[d], D. Robert[e]

[a] *Institute of Neuroinformatics, University/ETH Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland*
[b] *Institute of Zoology, University of Zürich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland*
[c] *Discipline of Physiology, School of Molecular & Biomedical Science and Centre for Biomedical Engineering (CMBE), University of Adelaide, Adelaide 5005, Australia*
[d] *Intelliact AG, Siewerdtstrasse 105, CH-8050 Zürich, Switzerland*
[e] *School of Biological Sciences, University of Bristol, Woodland Road, Bristol BS8 1UG, UK*

Received 11 September 2003; received in revised form 21 November 2003; accepted 12 December 2003

Abstract

The presentation of controllable, dynamic sensory stimuli provides a powerful experimental paradigm, which has been extensively applied to explore sensory processing in walking and tethered flying insects. Recent advances in computer hardware and software technology provide the opportunity to track the 3D flight path of free-flying insects and process these data in real-time, opening up the possibility to present dynamic stimuli to free-flying animals. To accommodate for the increased complexity relating to 3D space, we partitioned experimental design, real-time data acquisition and stimulus control into multiple self-contained modules. 3D experimental scenarios were created in a stand-alone application by forging multiple 3D space–stimulus relationships. The use of dynamic cues is illustrated by an experiment, in which dynamic acoustic cues were presented to a free-flying parasitoid fly in a large 3D environment. The combination of loosely coupled modules provides robust and flexible solutions, allowing new paradigms to be readily implemented based on existing technologies. We demonstrate this with a test system that displayed a complex visual stimulus, controlled in real-time by the 2D position and orientation of a test object. The presented methods are applicable in a variety of novel experimental paradigms, including learning paradigms, for various sensory modalities in walking, swimming and flying animals.
© 2004 Elsevier B.V. All rights reserved.

Keywords: 3D; Tracking; Behavior; Flight; Closed-loop; Vision; Acoustics; Insect

1. Introduction

Many flying insects perform aerial maneuvers that by far outclass man-made aircraft in rapidity and precision. As an example, flies perform abrupt turns (body saccades) within less than a tenth of a second, during which the body rotates at over one thousand degrees per second (Fry et al., 2003; Schilstra and Van Hateren, 1999). These maneuvers are part of a complex navigational strategy, in which the fly integrates visual and olfactory cues, e.g. to locate a food source (Frye and Dickinson, 2001; Frye et al., 2003). To cope with the navigational demands of fast and robust flight control, insects draw on multimodal afferent information about visual, olfactory (Frye and Dickinson, 2001) and acoustic (Cade, 1975; Robert and Rowell, 1992b; Walker and Wineriter, 1991) cues. Although an insect's nervous system weighs but a few milligrams, and typically contains fewer than 500,000 neurons, it operates with stunning efficiency and robustness. The high demands of flight have caused highly adapted physiological solutions to evolve, making insects powerful models for studying the accurate and fast processing of sensory information (Göpfert et al., 2002; Götz, 1968) and robust flight control (Fry et al., 2003; Taylor, 2001). Nature's solution to a variety of fundamental problems in sensory processing and motor control can provide important design principles for neuromorphic systems, such as Very Large Scale Integration (VLSI) hardware (Delbrück, 1993; Giles, 2001). Furthermore, a detailed understanding of the natural behavior of animals in 3D space provides an important basis for the design of biomimetic micro-robotic systems (Avadhanula et al., 2002; Schenato et al., 2002).

1.1. Dynamic sensory stimulation and behavioral context

In freely moving animals static objects are perceived under (natural) closed-loop conditions, in which the perceived

* Corresponding author. Tel.: +41-1-635-30-65; fax: +41-1-635-30-53.
E-mail address: steven@ini.phys.ethz.ch (S.N. Fry).

stimuli depend directly on the exerted motor behavior. Not only do animals react to these impinging stimuli (Wehner, 1981), they also influence them by their own motor behavior, in many cases actively and purposefully (Brünnert et al., 1994; Lehrer, 1991, 1993; Voss and Zeil, 1998; Zeil, 1997). To understand how animals process sensory information, it is therefore important to determine complex stimulus–response relationships, a task which is often unfeasible under field conditions (but see: Collett and Land, 1975; Land and Collett, 1974; review: Wehner, 1992). In contrast to field experiments, experiments using tethered animals in the laboratory allow stimulus–response relationships to be explored under open-loop conditions, in which the animal has no influence over the perceived auditory (Nolen and Hoy, 1986; Robert, 1989) or visual (Götz, 1964, 1968; Heide and Götz, 1996; Reichardt and Wenking, 1969) stimulus. To allow for more naturalistic stimulus conditions under tethered conditions, closed-loop stimulus conditions are artificially generated in the classic 'flight simulators'. Here, a fly's intention to turn is measured with a yaw torque meter or wing beat analyzer (Fig. 1A), and used to control the azimuth of a visual target in real-time (Götz, 1964, 1968; Heisenberg and Wolf, 1984). These intricate experimental setups extend far beyond merely attempting to simulate free-flight conditions, but instead allow an animal's complex behavior to be explored by dynamically controlling the sensory input from an animal's behavior in a manner suitable to answer the specific questions (Dickinson and Lighton, 1995; Lehmann and Dickinson, 1997; Robert and Rowell, 1992b; Wolf et al., 1992). As an example, Tammero and Dickinson (2002) have stimulated a tethered flying fly with a visual target, of which the azimuth was controlled in closed-loop, but the size was controlled in open loop. In this way, the fly was caused to actively fixate an object in its frontal visual field and to perform an object-avoidance maneuver when the object was suddenly expanded.

Flight simulators cannot replace experiments with freely moving animals, however. First, sensory stimulation is typically provided only in a single sensory modality (but see Frye and Dickinson, 2003; Sherman and Dickinson, 2003). Unlike in free flight, mechanosensory feedback from the halteres or other equilibrium sensors is lacking, which causes the flight behavior to differ from that of free flight (Fry et al., 2003; Fry et al., in preperation; Mayer et al., 1988). Second, important information is lost because large constraints are imposed on the insect's behavioral repertoire. In most flight simulators only a single degree of freedom (e.g. yaw torque) is measured out of a total of six in free flight. Third, tethering is likely to cause behavioral artifacts due to the mechanical interference and sensory distortions caused by the tether (Robert and Rowell, 1992a). Taken together, dynamic stimulus presentation in tethered animals provides a powerful experimental paradigm that is, however, limited due to the requirement of tethering the animal.

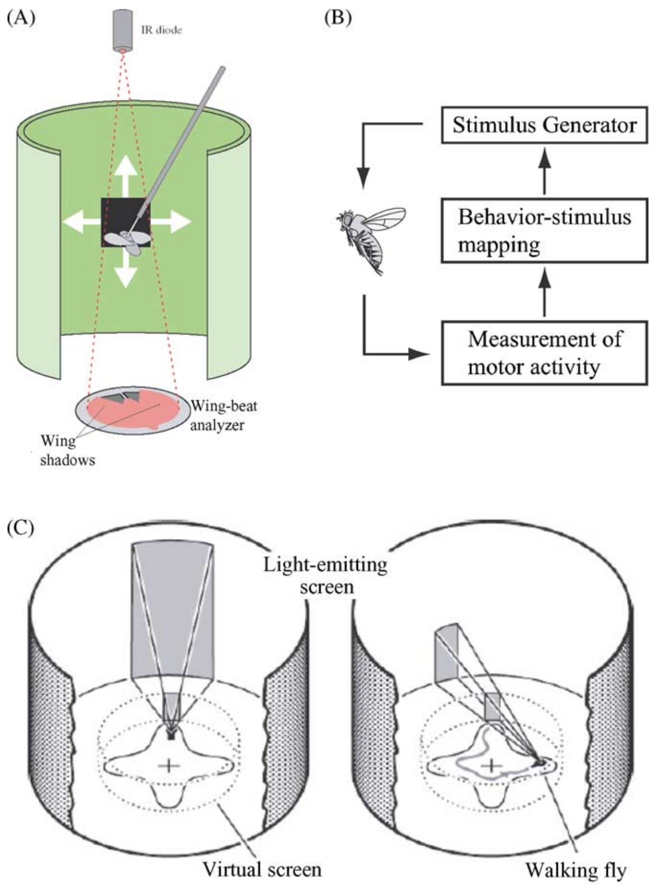


Fig. 1. Examples of methods allowing dynamic stimulus manipulation in flies. (A) Flight simulator. A wing beat analyzer (WBA) measures the difference in wing beat amplitude between the wings, and hence the fly's intention to turn. This signal is then used to control the position of an object (black square) displayed on a panoramic LED screen in closed-loop. The size of the object is controlled in open loop, such that it can be caused to suddenly loom. Figure from Tammero and Dickinson (2002). (B) Schematic of the components comprising a system creating dynamic stimuli. (1) A device to measure the animal's motor output in real-time. (2) A controller for generating the specific stimulus parameters based on the animal's current behavior. (3) A stimulation device. (C) Dynamic visual stimulation of a fruit fly walking within a panoramic LED display. The fly's position and body axis direction were measured with a charge-coupled device (CCD) camera and used to generate a virtual object that changed in position and size according to the fly's behavior ('virtual open-loop'). Figure from Schuster et al. (2002).

1.2. Questions and goals

Irrespective of the specific implementation, a system that allows dynamic stimulus presentation must consist of at least the following components (Fig. 1B): First, a device to measure the motor output of the animal in real-time (torque meter, wing beat analyzer, video camera, etc.). Second, a signal processor to generate the parameters used for sensory stimulation (e.g. analog hardware, computer). Third, a device to present the sensory stimuli. The opportunity to measure the 3D flight path of free-flying flies in real-time opens up the opportunity to apply dynamic stimulus control in free-flying animals. That dynamic stimulus presentation needs not be restricted to tethered animals has been elegantly demonstrated by Schuster et al. (2002), who presented virtual visual objects to fruit flies walking within a panoramic LED display (Fig. 1C). We took this approach further by developing methods that allowed complex experimental scenarios to be planned and used to present dynamic—in our case acoustic—stimuli to free-flying flies in a large 3D environment. For this, 3D position data were acquired using Trackit 3D, a real-time path acquisition system described in detail elsewhere (Fry et al., 2000). Next, we needed a method that allowed us to construct complex and flexible experimental scenarios. Mathematical descriptors soon become unintuitive if applied to 3D space, and therefore a graphical representation is often more suitable. The methods developed to plan experiments in a 3D space, represented intuitively and unambiguously on a 2D display, are described in Section 2.2. To provide a flexible and robust solution, we chose a modular approach, in which experimental design, data acquisition and stimulus presentation were organized in self-contained modules. One such module was dedicated to the design of 3D experimental scenarios, which were then used in a separate module to present dynamic acoustic stimuli to a free-flying fly during experiments (Section 2). The benefits of this approach and its potential in similar experimental paradigms is then demonstrated with a system, in which real-time tracking and a software module based on an open-source graphics library were combined to dynamically control a visual stimulus (Section 3).

2. Dynamic acoustic stimulation in a 3D environment

Females of the parasitoid fly species *Ormia ochracea* locate their host, a cricket, by means of acoustic cues. Even in pitch dark, the fly is able to recognize a singing cricket, fly toward it and land in its close proximity to larviposit. To investigate the mechanisms underlying host detection and localization in *Ormia*, the host-seeking behavior can be elicited under controlled laboratory conditions (Müller and Robert, 2001; Ramsauer and Robert, 2000). Flies placed on a small platform remain inactive until a synthesized sound of a singing cricket is played from a loudspeaker located on the floor of the flight room. Within seconds after the onset of the sound, the fly reaches the active loudspeaker. In reality, the acoustic environment is more complex, in that the songs produced by the crickets are discontinuous. Furthermore, in the fly's natural habitat, several singing crickets are located at different places (Fig. 2A). The fly is thus confronted with the complex task of picking an individual cricket out of a population of intermittently singing individuals. To explore how *Ormia* performs under such challenging conditions, we tested it with a suite of scenarios that served as realistic duplicates of natural situations. A simple example consisted of a (virtual) cricket that ceased to sing as it was being approached by the fly, while at the same time a different cricket began to sing in a different location. The technical implementation of the 'virtual field experiments' under laboratory conditions are shown in Fig. 2B, the details of which are described in the following sections.

2.1. Modular design for dynamic stimulus presentation in 3D

Rather than implementing an integrated system to measure the flight path and generate acoustic stimuli, we approached the problem by solving various tasks in separate linked modules. Constructing various experimental scenarios in 3D space is a delicate problem, for which we developed self-contained software that allowed experiments to be designed and the scenarios to be stored on the computer's

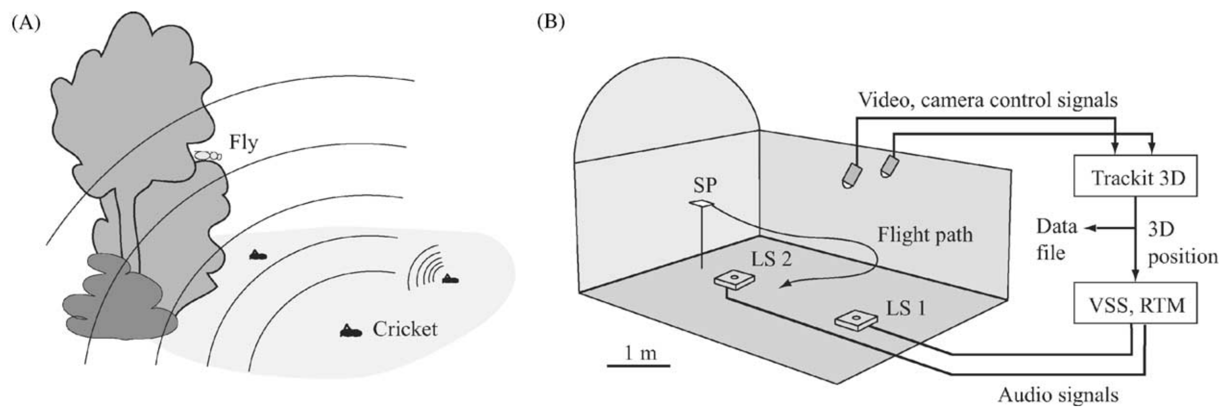


Fig. 2. Dynamic stimulus conditions experienced by the parasitic fly *O. ochracea* in the field and in the laboratory. (A) Possible scenario *O. ochracea* is faced with in the field. The fly is located in a shrub next to a meadow with several calling crickets. The fly takes off and approaches a singing cricket, which ceases to sing, while a different cricket begins to sing in a different location. (B) Laboratory setup allowing dynamic acoustic stimuli to be presented to a free-flying parasitoid fly. Individual flies were released from the start platform (SP) and their 3D position was tracked in real-time using Trackit 3D, equipped with two pan-tilt cameras. The position data were streamed to the RTM (2.3), which controlled the audio output in near real-time, according to the pre-defined experimental scenario (VSS programmer, 2.2). Sound output was from two loudspeakers located on the floor (LS1 and LS2).

hard disk (Virtual Stimulus Space Programmer (VSS Programmer), Section 2.2). To perform an experiment, a stored VSS Programmer scenario was loaded into the 'real-time module' (RTM, Section 2.3), itself a custom-developed stand-alone application. It controlled generation of the sensory stimuli, in our case various samples of artificial cricket songs, in response to real-time data input from Trackit 3D (Section 2.4). Output of the RTM was to a sound card, the audio output of which was amplified and used to drive a pair of loudspeakers located on the floor of the flight chamber.

At the beginning of an experiment, a fly was placed on the starting platform (SP) in the flight chamber (Fig. 2B). Position tracking in Trackit 3D was activated and then the RTM started. In one scenario (Fig. 4A), sound was emitted from the more distant of a pair of loudspeakers (LS1 and LS2). The fly typically took to the wing within a few seconds and began its approach toward the loudspeaker. The fly's position was continuously displayed in the RTM's graphical user interface (GUI), allowing the experiment to be monitored from a remote location. As the fly crossed a plane (shown in purple in Fig. 4A) between the two loudspeakers, the RTM shut off the sound emission from LS1 and played a short sound pulse from LS2. After the fly had landed, tracking was terminated and it was removed from the flight chamber. Trackit automatically stored a file containing the sampled 3D positions of the fly on the computer's hard disk. In this way, a large number of successive experiments could be performed efficiently.

2.2. *The VSS Programmer, a module for designing intricate 3D experimental scenarios*

Technically, the task mentioned above was as follows: "Play sound A in position 1, until the fly reaches position X, then terminate sound A and play sound B in position 2". For the specific example, this required various sound parameters to be controlled, such as duration of the sound pulse and the positions where sounds were switched on or off. To be able to perform a large variety of different experiments, we needed a general method that allowed scenarios to be programmed flexibly and efficiently. In this more general form, the task took on the following form: "Depending on the location of the fly, play a specific sound sequence from a specific location". Ultimately, these associations must be implemented algorithmically in the program code, allowing action to be taken depending on the fly's measured position. Direct coding of 3D space–stimulus relationships lacks intuition and is highly error-prone, and was therefore dismissed. Instead, we implemented an interactive GUI, the Virtual Stimulus Space Programmer, using LabView V 5.1 (National Instruments), running on an Apple Macintosh PPC 7600; 96 MB RAM, 133 MHz. Stimulus control was performed within a cuboid region of the virtual stimulus space, the 'control space' (CS, Fig. 4A). The application's GUI displayed projections of the CS onto three orthogonal planes (Fig. 3A). This redundant representation (total of six coordinates for 3 *df*) of the CS in the display was chosen as a means to define multiple, overlapping and non-overlapping 3D objects (also see Fry et al., 2003). Two basic types of objects could be generated, cuboids and spheres (Fig. 3B), represented by circles and rectangles in the two-dimensional projections (Fig. 3A), respectively. Size and position of each of these objects could be manipulated in the GUI, either using the computer mouse or by numeric input of coordinates using the keyboard (defining side lengths and diameters of the cuboids and spheres, respectively). Any changes caused an instant refresh of the displayed objects and coordinate values.

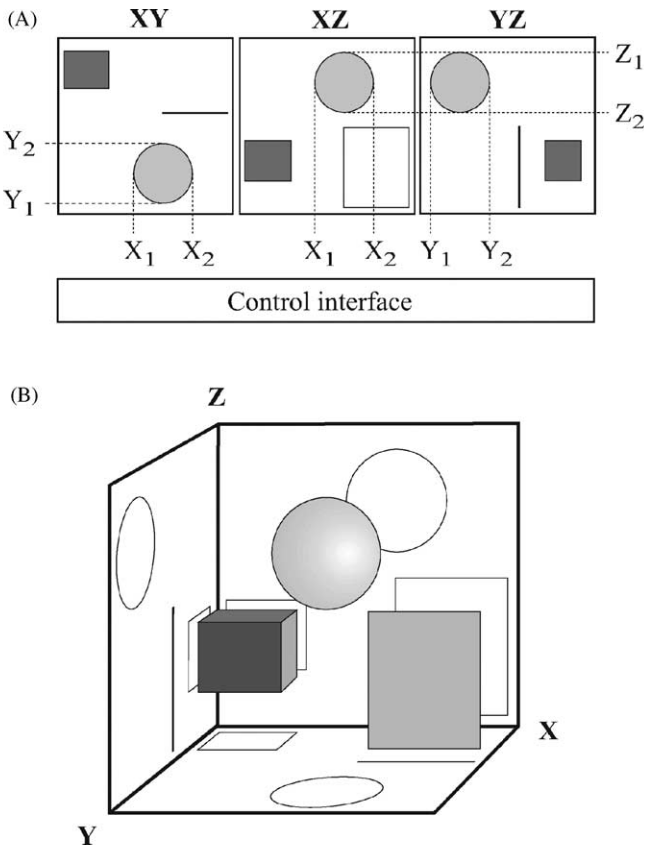


Fig. 3. 2D representation of the 3D control space. (A) Control interface of the VSS Programmer. Three planar projections of the 3D control space were displayed in the VSS Programmer's GUI. A horizontal projection (*XY*, also see notation in Fig. 4A) and two vertical projections (*XZ* and *YZ*) were used to represent the control space. Virtual cuboids and spheres could thus be generated within the CS, and linked with previously sampled sound files in an intuitive way. The spatial attributes of the 3D objects could be changed by dragging the object on the screen using the computer's mouse, as well as by editing coordinate values displayed in the GUI. (B) Perspective view of the 3D control space, as defined in the GUI shown in A.

Each object thus defined was next associated with a stored sound sample, which had previously been sampled at 44 kHz stereo using SoundEdit16. In this way the content, duration and intensity of the sound output was defined in a simple way. The right and left audio channels were used to choose the loudspeaker (LS1 or LS2, Fig. 2B). The experimental scenario thus created was stored on hard disk as an ASCII text file. It contained a table consisting of as many rows as there were defined 3D objects, and 12 columns of the format: O_i, C_i, X_1, Y_1, Z_1, X_2, Y_2, Z_2, S_i, V_L, V_R, OT, where O_i, C_i, S_i are indices of an object, its screen color and the sound file associated with it, respectively. X_1, Y_1, Z_1 and X_2, Y_2, Z_2 define a 3D vector representing the diagonal of a cuboid. V_R, V_L contains the volume settings for the right and left audio channel, respectively. OT is a switch that defines whether the object is a cuboid of the specified dimensions, or else a sphere. In the latter case the position of a cube bounding the sphere is specified.

2.3. The real-time module (RTM)

Software for stimulus production, the RTM, was developed in the same programming environment as the VSS Programmer. The user could load the stored scenario into the computer's memory. The RTM provided a GUI displaying the VSS very much the same way as the VSS Programmer. A test mode allowed the user to perform a 'dry run' of the experiment. Instead of using live position measurements of a flying fly, the position of a 'virtual' fly was manually controlled by moving a cursor on the screen. The RTM used this position to trigger the appropriate sound stimuli as specified in the 3D scenario. This feature proved indispensable for verifying the correct production of the stimuli. Furthermore, it provided a simple interface to switch between various stimuli, which simplified calibration measurements of the stimulus delivered, such as the sound pressure level.

To perform actual experiments, real-time data input from Trackit 3D (Section 2.4) was used to generate the appropriate sound stimuli. During an on-going experiment the RTM displayed the currently measured position of the fly, such that the experimental progress could be monitored from a remote location.

2.4. Position measurement and 3D position data transfer

We measured the fly's instantaneous position using Trackit 3D, running on Windows NT (Microsoft), configured with two pan-tilt cameras (Sony LSX-PT1, custom modified for infra-red sensitivity). Lighting was provided by custom-built IR lamps (10×10 arrays of infra-red light emitting diodes, TSMA 6503 LED, $I_F \sim 100$ mA, $\lambda = 875$ nm, running on a dc power source). The lighting provided for the filming did not compromise the flies' natural behavior, as they are insensitive to light in this range (Stark and Johnson, 1980). The fly's 3D position was sampled at 50 Hz (corresponding to the interlaced refresh rate of standard PAL video equipment), converted to a suitable binary format (see below) and streamed to the computer's parallel port. The fly's measured 3D position, P_{meas}, was mapped to the position in the control space, P_{CS}, as:

$$P_{\text{CS}} = \left| \frac{P_{\text{meas}} - P_{\min}}{P_{\max} - P_{\min}} \right| \times 256, \; P_{\text{CS}} \in (P_{\min}, P_{\max})$$

$$P_{\text{CS}} = 0, \; P_{\text{meas}} \leq P_{\min}$$

$$P_{\text{CS}} = 255, \; P_{\text{meas}} \geq P_{\min},$$

with $P_{\min}$ and $P_{\max}$ the lower and upper limits, respectively, in each dimension. Put in words, the coordinates measured by Trackit 3D within a pre-defined volume were mapped onto values between 0 and 255, or 8 bit, in each dimension. In total, 32 bits (Header, X, Y, Z) were streamed to the computer's parallel port at 50 Hz. As a serial protocol was preferred for the input to the RTM, we converted the parallel data stream to a 19200 baud serial protocol using standard parallel-serial converter hardware. Depending on the specific implementation used, a parallel interface, a network interface (Section 3.3), or other reliable methods to stream data between computers may serve the same function.

3. Dynamic visual stimulus presentation

The methods allowing the presentation for dynamic acoustic stimuli to free-flying flies can likewise be applied for various experimental scenarios in flying, walking and swimming animals. Applying similar methods for presenting dynamic visual stimuli to unrestrained moving animals is of high relevance, as most of the previous research in insects has been performed using visual stimuli. There are, however, technical difficulties relating to the presentation of dynamic visual stimuli, due to the technical limitations of standard displays. The typical refresh rates of the cathode ray tube (CRT) displays traditionally used as computer monitors are around 90–120 Hz, which is well above the critical fusion frequency in humans, which lies around 60 Hz (Landis, 1954). On the contrary, these frequencies are easily perceived by some insects, such as blowflies, which may be able to sense flicker up to 400 Hz or 500 Hz (Tatler et al., 2000). Liquid crystal displays (LCD) are not burdened with flicker, but instead have a limited refresh frequency of around 50 Hz, which can lead to 'ghosting' effects for objects in motion (Straw and O'Carroll, 2003). Finally, daylight luminance levels and natural amounts of contrast are difficult to produce in the laboratory, irrespective of the technology used. Maximal performance in terms of luminance contrast and refresh rate are often achieved with custom-designed stimulation devices. For instance, Lindemann et al. (2003) have constructed a 370 Hz panoramic display ('FliMax'), based on custom LED panels and sample-and-hold electronics that eliminate flicker and produce a maximum luminance of about 400 cd/m^2. As FliMax receives image data from a standard VGA card, it could easily be modified to generate real-time visual stimuli from a computer system (Jens Peter Lindemann, personal communication). Therefore, the existing methods allowing the design of high-performance displays could be combined with our methods to create dynamic visual stimuli for other experimental contexts.

3.1. Dynamic visual stimulus presentation

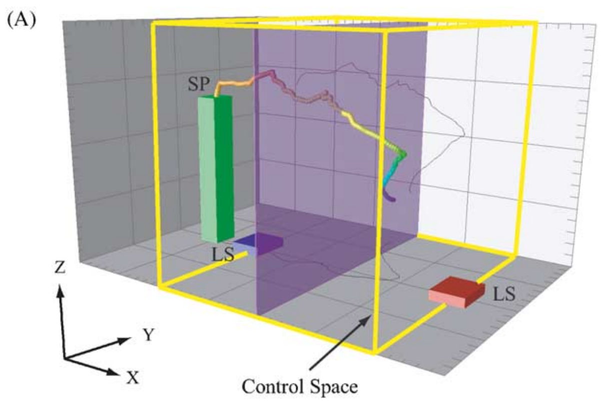
In spite of the abovementioned complications, there are many applications with lesser requirements for visual stimulation, allowing for simpler solutions, in particular if fast object motion and high refresh frequencies are not required. In that case, conventional computer displays can provide a useful alternative to custom-built hardware solutions (Bach et al., 1997; Cowan, 1995; Packer et al., 2001). That standard technology can be combined with our methods to provide dynamic visual stimulation, and moreover that this can be accomplished easily and cheaply, is demonstrated with a system that we used to test the feasibility of dynamic visual stimulation for experiments planned with walking, and later flying, fruit flies (*Drosophila*). Because fruit flies are unable to sense flicker above 200 Hz (Juusola and Hardie, 2001), we chose a standard CRT computer monitor that may be updated at that rate (Flatron 915 FT+, LG Electronics). Preliminary experiments have shown that measuring position and body axis direction in 2D, as well as 3D position, yaw and pitch angle, is reliably possible in this species using Trackit 2D/3D. For reasons of simplicity, we used two-dimensional position and body axis data acquired in real-time using Trackit 2D as input for a modular visual stimulator.

3.2. Real-time visual stimulus presentation using the 'Vision Egg'

We generated and displayed visual stimuli using the 'Vision Egg', a Python programming library that makes use of the Python OpenGL interface (PyOpenGL). A detailed description of the VisionEgg lies beyond the scope of this report and will be presented elsewhere (Straw, in preparation). The open source code and documentation are available on the Internet (http://www.visionegg.org). We used VisionEgg Version 0.9.4 and Python 2.2 on a 600 MHz Pentium III PC running Windows 2000 Professional and equipped with an nVidia GeForce4 MX 420 graphics card.

For the present tests, we replaced the fly with a 2D test object, a black stripe that was moved manually over of a bright background. The 2D position and orientation of the stripe were transferred from Trackit in real-time using a network socket connection, which allows for added flexibility (Section 3.3). Depending on the test object's momentary position and orientation, a 120° section of a virtual reality (VR) panorama was presented in 'virtual open loop' (Schuster et al., 2002) (Fig. 4B, also see Fig. 1C). Moving the test object closer to the screen caused the image to shrink away, such that a constant angular size of the displayed objects was maintained. Similarly, rotation of the stripe was followed by the pattern without noticeable delay. Although perspective distortion is computationally demanding, the image was rendered at 200 Hz without skipping frames, while accepting real-time input from Trackit. Because OpenGL allows the computationally demanding rendering to be performed on the graphics card at 200 Hz, the (moderately fast) CPU was largely free to do other tasks, such as check for new data from Trackit.

To quantify the latency of the system, we configured the software to display a small test object on the screen. Its position, as measured by Trackit and transferred via the network interface, was logged together with the test object's position. The software caused the test object to 'jump' to a



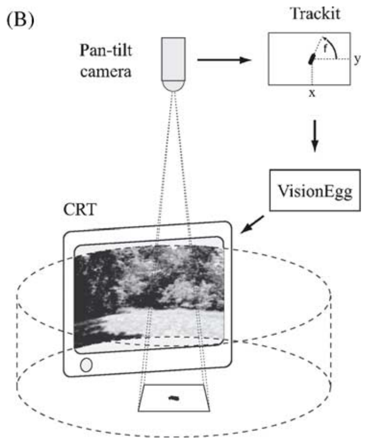


Fig. 4. Dynamic stimulation using acoustic and visual stimuli. (A) Example of an experimental scenario used to explore acoustic host-seeking behavior in a parasitoid fly. A continuous sound was emitted from the loudspeaker (LS, shown in red), causing the fly to take off from the starting platform and approach the loudspeaker. As soon as the fly crossed the purple plane in the control space (CS, yellow wire frame), the sound was shut off and a brief sound sample consisting of five pulses was emitted from the loudspeaker closer to the starting point, shown in blue. The recorded flight trajectory illustrates the ability of the fly to react and orient toward a transient sound source. A detailed description of this and other experiments performed using the same setup is given in Müller and Robert (2001). (B) Test setup used to generate dynamically controlled visual stimuli. The 2D position and orientation of a test object (black stripe) was measured using Trackit 2D and streamed to the VisionEgg, which used this information to display a 120° section of a virtual reality (VR) panorama in 'virtual open loop' (Schuster et al., 2002, also compare with Fig. 1C). The image was rendered at 200 Hz without skipping frames, while reacting to real-time input from Trackit.

new location and measured the time that passed before the new position was registered. The total latency of the combined system was around 60 ms. A large part of the time lag is explained by the relatively low sampling rate of the PAL video hardware used (operating at 50 Hz). The use of faster digital cameras that allow images to be acquired at up to 200 Hz using standard frame grabbers could significantly reduce the time lag. If moderate refresh rates and a time lag of 60 ms are tolerable, our methods provide the possibility to present dynamic visual stimuli using standard, and hence affordable, technology.

3.3. Data transfer using a network socket connection

As an alternative method to the serial data transfer described in Section 2.4, we implemented a network connection, allowing ASCII data to be streamed from Trackit to the vision module making use of the standard network hardware included in computer hardware (TCP/IP). For this, the computers were connected with a crossover networking cable. Trackit acted as the server application, creating an open socket and listening for a connection from a remote program. To allow external software to access the position data, a small ‘Trackit client library’ was implemented in C. For each acquired frame, Trackit pushes the measured coordinates (e.g. position and body axis direction) onto a stack. The library provides two functions that allow data to be retrieved from the stack. Repeated calls of ‘getNextMessageFromSocket( )’ sequentially retrieve position data from the stack. The data returned by the function are of the form: “X Y x y”. X and Y specify the horizontal and vertical position, respectively, of the centroid of the object in camera coordinates. Units are degrees, zero denoting the image center. x and y represent the cosine and sine, respectively, of the body angle. The body angle θ can be calculated from $\theta = \arctan(y/x)$, $\theta \in [-90^\circ, 90^\circ]$.

As is apparent from the formula, the body axis direction is ambiguous (360° are mapped onto 180°), and additional algorithms are required to reconstruct the complete 360° orientation, based on appropriate assumptions about the body dynamics.

While this method guarantees that no position data are lost, a backlog might occur if data retrieval were slower than the sampling rate of Trackit (50 Hz PAL/60 Hz NTSC). For this situation an additional function is provided, ‘getLastMessageFromSocket( )’, which retrieves the last, most recent item from the stack and discards all older data.

The Trackit library functions were encapsulated in a small module written in C that allowed them to be called directly from Python. This method allowed us to query Trackit in real-time over the network, without the need to perform network functions explicitly. The described methods are not limited to Python, but can likewise be applied to a large variety of external programming languages capable of interfacing with C, including C, Perl, Matlab, LabView, and many more.

4. Discussion

Real-time control of sensory stimuli has proven to be an extremely powerful experimental paradigm, in particular for the presentation of visual stimuli to tethered flying insects. Due to the high demands for processing speed, custom-build hardware solutions are often required for measuring an animal’s response and controlling a dynamic visual stimulus in real-time. Rapid advances in the electronics industry provide the additional option to acquire behavioral data and process it in real-time using digital technology and software solutions (review: Reynolds and Riley, 2002). In particular, the possibility of acquiring the 3D position of a free-flying animal in real-time and using it to control sensory stimuli in real-time allows new experimental paradigms to be created.

The design of complex dynamic stimulation experiments in a 3D environment is no trivial problem, however, because 3D space cannot be directly represented on a two-dimensional medium, such as a display. Furthermore, the additional complexity of real-time 3D stimulus control runs counter to a flexible and manageable solution. We tackled these problems by applying a rigorously modular approach that allowed even complex 3D experiments to be planned and carried out using flexible and intuitive tools. One module, the VSS Programmer, was used to design 3D experimental scenarios using an intuitive graphic user interface. The programmed scenarios were then tested in a separate module and applied in experiments using real-time position input from an independent path acquisition system (Trackit). The successful implementation demonstrates that complex experimental paradigms, including dynamic stimulation in 3D space, no longer need be limited to slow or immobilized animals.

A further advantage of modularization is the ability to combine various subsystems to create new implementations, without having to ‘re-invent the wheel’ for each new application. For instance, instead of presenting acoustic stimuli to a flying fly, the system can be adapted to present visual stimuli to a honeybee visiting a feeder, to which is has previously been trained. To demonstrate that dynamic visual stimulation based on the same design principles is indeed possible, albeit within limits, we implemented a test system to generate complex dynamic visual stimuli. For this, we combined the VisionEgg and Trackit 2D, which allowed us to generate a ‘virtual open-loop’ stimulus at a refresh rate of 200 Hz, which reacted to the position and orientation of a test object with a time lag of 60 ms.

While the methods and design principles presented in this report cannot—and do not intend to—provide a custom solution for a single problem, they do provide the methods and principles allowing sophisticated dynamic stimulation paradigms to be created, in particular for 3D environments. Also, further powerful extensions of the application are apparent, such as the potential of using dynamic stimuli in sophisticated training paradigms. Taking the example with the honey bee a step further, our methods can easily be extended to create complex, and fully automated learning paradigms, in which the bee's flight behavior is recorded and used to manipulate the appearance of static and moving visual stimuli. With an increasing need to understand the biophysical functioning and decision-making mechanisms of animals in realistic behavioral tasks, automated and 'smart' experimental paradigms promise powerful new tools for behavioral neuroscience.

Acknowledgements

We wish to thank H. Heise for building various pieces of equipment used in the flight chamber, H. Kohler for rearing the flies, M.H. Dickinson for useful comments on the manuscript and A.M. Whatley for proof-reading. Two reviewers have provided useful suggestions. The work was supported by the Swiss National Science Foundation, the Claraz Foundation, the Schering Foundation, the University of Bristol and the Howard Hughes Medical Institute.

References

Avadhanula S, Wood RJ, Campolo D, Fearing RS. Dynamically tuned design of the MFI thorax. IEEE International Conference on Robotics and Automation; 2002.

Bach M, Meigen T, Strasburger H. Raster-scan cathode-ray tubes for vision research. Limits of resolution in space, time and intensity, and some solutions. Spat Vis 1997;10:403–14.

Brünnert U, Kelber A, Zeil J. Ground-nesting bees determine the location of their nest relative to a landmark by other than angular size cues. J Comp Physiol A 1994;175:363–9.

Cade Q. Acoustically orienting parasitoids: fly phonotaxis to cricket song. Science 1975;190:1312–3.

Collett TS, Land MF. Visual control of flight behaviour in the hoverfly *Syritta pipiens* L. J Comp Physiol 1975;99:1–66.

Cowan WB. Displays for vision research. In: Bass M (Ed.). Handbook of optics, vol. 1, Fundamentals, techniques, and design. New York: McGraw-Hill; 1995. p. 27.1–27.44.

Delbrück T. Silicon retina with correlation-based, velocity-tuned pixels. IEEE Trans Neural Networks 1993;4:529–41.

Dickinson MH, Lighton JRB. Muscle efficiency and elastic storage in the flight motor of *Drosophila*. Science 1995;268:87–90.

Fry SN, Bichsel M, Müller P, Robert D. Tracking of flying insects using pan-tilt cameras. J Neurosci Methods 2000;101:59–67.

Fry SN, Sayaman R, Dickinson MH. The aerodynamics of free-flight maneuvers in *Drosophila*. Science 2003;300:495–8.

Frye MA, Dickinson MH. Fly flight: a model for the neural control of complex behavior. Neuron 2001;32:385–8.

Frye MA, Dickinson MH. A signature of salience in the *Drosophila* brain. Nat Neurosci 2003;6:544–6.

Frye MA, Tarsitano M, Dickinson MH. Odor localization requires visual feedback during free flight in *Drosophila melanogaster*. J Exp Biol 2003;206:843–55.

Giles J. Think like a bee. Nature 2001;410:510–2.

Göpfert MC, Stocker H, Robert D. *Drosophila* atonal is required for the formation of both the neural and exoskeletal components that transform fly antennae into ears. Zoology (Jena) 2002;105:77.

Götz KG. Optomotorische Untersuchung des visuellen Systems einiger Augenmutanten der Fruchtfliege *Drosophila*. Kybernetik 1964;2:77–92.

Götz KG. Flight control in *Drosophila* by visual perception of motion. Kybernetik 1968;4:199–208.

Heide G, Götz KG. Optomotor control of course and altitude in *Drosophila melanogaster* is correlated with distinct activities of at least three pairs of flight steering muscles. J Exp Biol 1996;199:1711–26.

Heisenberg M, Wolf R. Vision in *Drosophila*: genetics of microbehavior. Berlin, Heidelberg, New York, Tokyo: Springer; 1984.

Juusola M, Hardie RC. Light adaptation in *Drosophila* photoreceptors. I. Response dynamics and signaling efficiency at 25 °C. J Gen Physiol 2001;117:3–25.

Land MF, Collett TS. Chasing behaviour of houseflies (*Fannia canicularis*). J Comp Physiol A 1974;89:331–57.

Landis C. Determinants of the critical flicker-fusion threshold. Physiol Rev 1954;34:259–86.

Lehmann FO, Dickinson MH. The changes in power requirements and muscle efficiency during elevated force production in the fruit fly *Drosophila melanogaster*. J Exp Biol 1997;200:1133–43.

Lehrer M. Locomotion does more than bring the bee to new places. In: Goodman LJ, Fisher RC (Eds.). The behaviour and physiology of bees. Wallingford, England, UK: C.A.B. International; 1991. p. 185–202.

Lehrer M. Why do bees turn back and look? J Comp Physiol A 1993;172:549–63.

Lindemann JP, Kern R, Michaelis C, Meyer P, Van Hateren JH, Egelhaaf M. *FliMax*, a novel stimulus device for panoramic and high-speed presentation of behaviourally generated optic flow. Vision Res 2003;43:779–91.

Mayer M, Vogtmann K, Bausenwein B, Wolf R, Heisenberg M. Flight control during 'free yaw turns' in *Drosophila melanogaster*. J Comp Physiol A 1988;163:389–99.

Müller P, Robert R. A shot in the dark: the silent quest of a free-flying phonotactic fly. J Exp Biol 2001;204:1039–52.

Nolen TG, Hoy RR. Phonotaxis in flying crickets. I. Attraction to the calling song and avoidance of bat-like ultrasound are discrete behaviours. J Comp Physiol A 1986;159:423–39.

Packer O, Diller LC, Verweij J, Lee BB, Pokorny J, Williams DR, Dacey DM, Brainard DH. Characterization and use of a digital light projector for vision research. Vision Res 2001;41:427–39.

Ramsauer N, Robert D. Free-flight phonotaxis in a parasitoid fly: Behavioural thresholds, relative attraction and susceptibility to noise. Naturwissenschaften 2000;87:315–9.

Reichardt W, Wenking H. Optical detection and fixation of objects by fixed flying flies. Naturwissenschaften 1969;56:424–5.

Reynolds DR, Riley JR. Remote-sensing, telemetric and computer-based technologies for investigating insect movement: a survey of existing and potential techniques. Comput Electron Agric 2002;35:271–307.

Robert D. The auditory behaviour of flying locusts. J Exp Biol 1989;147:279–302.

Robert D, Rowell CHF. Locust flight steering. I. Head movements and the organization of correctional manoeuvres. J Comp Physiol A 1992a;171:41–51.

Robert D, Rowell CHF. Locust flight steering. II. Acoustic avoidance manoeuvres and associated head movements compared with correctional steering. J Comp Physiol A 1992b;171:53–62.

Schenato L, Deng X, Sastry S. Hovering flight for a micromechanical flying insect: modeling and robust control synthesis. In: 15th IFAC World Congress on Automatic Control; 2002.

Schilstra C, Van Hateren JH. Blowfly flight and optic flow. I. Thorax kinematics and flight dynamics. J Exp Biol 1999;202:1481–90.

Schuster S, Strauss R, Götz KG. Virtual-reality techniques resolve the visual cues used by fruit flies to evaluate object distances. Curr Biol 2002;12:1591–4.

Sherman A, Dickinson MH. A comparison of visual and haltere-mediated equilibrium reflexes in the fruit fly *Drosophila melanogaster*. J Exp Biol 2003;206:295–302.

Stark WS, Johnson M. Microspectrophotometry of *Drosophila* visual pigments: determinations of conversion efficiency in R1-6 receptors. J Comp Physiol 1980;140:275–86.

Straw AD, O'Carroll DC. Motion blur applied to eliminate artifacts in apparent motion displays (Abstract, Vision Science Society 2003 Annual Meeting) J Vis 3;2003:782a.

Tammero LF, Dickinson MH. Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly, *Drosophila melanogaster*. J Exp Biol 2002;205:2758–98.

Tatler B, O'Carroll DC, Laughlin SB. Temperature and the temporal resolving power of fly photoreceptors. J Comp Physiol A 2000;186:399–407.

Taylor GK. Mechanics and aerodynamics of insect flight control. Biol Rev Cambridge Philos Soc 2001;76:449–71.

Voss R, Zeil J. Active vision in insects: an analysis of object-directed zig-zag flights in wasps (*Odynerus spinipes*, Eumenidae). J Comp Physiol A 1998;182:377–87.

Walker TJ, Wineriter SA. Hosts of a phonotactic parasitoid and levels of parasitism (Diptera: Tachinidae: *Ormia ochracea*). Fla Entomol 1991;74:554–9.

Wehner R. Spatial vision in arthropods, Handbook of sensory physiology. VII/6C. Berlin, Heidelberg, New York, Tokyo: Springer; 1981. p. 287–616.

Wehner R. Arthropods. In: Papi F, editor. Animal homing. Chapman & Hall; 1992. p. 45–144.

Wolf R, Voss A, Hein S, Heisenberg M. Can a fly ride a bicycle? Philos Trans R Soc Lond, Ser B: Biol Sci 1992;337:261–9.

Zeil J. The control of optic flow during learning flights. J Comp Physiol A 1997;180:25–37.