Virtual Reality Platform for Systematic Investigation of Multisensory Integration and Training of Closed-Loop Prosthetic Control

Kris Phataraphruk, Paul VanGilder, Christopher A Buneo

Abstract— **Multisensory integration is the process by which information from different sensory modalities is integrated by the nervous system. Understanding this process is important not only from a basic science perspective but also for translational reasons, e.g. for the development of closed-loop neural prosthetic systems. Here we describe a versatile virtual reality platform which can be used to study the neural mechanisms of multisensory integration for the upper limb and could potentially be incorporated into systems for training of robust neural prosthetic control. The platform involves the interaction of multiple computers and programs and allows for selection of different avatar arms and for modification of a selected arm's visual properties. The system was tested with two non-human primates (NHP) that were trained to reach to multiple targets on a tabletop. Reliability of arm visual feedback was altered by applying different levels of blurring to the arm. In addition, tactile feedback was altered by adding or removing physical targets from the environment. We observed differences in movement endpoint distributions that varied between animals and visual feedback conditions, as well as across targets. The results indicate that the system can be used to study multisensory integration in a well-controlled manner.**

I. INTRODUCTION

Making reaching movements and interacting with objects in our environment requires an estimate of the postural configuration of the limb (i.e. body schema) and its location relative to objects of interest. Inherent noise in the nervous system can create uncertainty in these state estimates [1], though this uncertainty can be reduced by combining sensory information (vision, proprioception, tactile) through a process called multisensory integration [2]. A common computational framework posits that sensory cues are integrated in a Bayesian manner based on their relative reliability – reliability which can change depending on sensory and/or motor context [3]–[5]. The neural mechanisms underlying this phenomenon are still relatively unknown, but are important for understanding sensorimotor processing [6], [7], and may have translational applications for neural prosthetic or sensory substitution systems. We have developed a platform that pairs a realistic virtual reality (VR) environment with motion tracking of the limb, enabling us to manipulate sensory (visual) information in real-time while subjects interact with virtual objects. By incorporating additional VR "games" into the system, we can systematically investigate the neural correlates of multisensory integration during volitional limb movements. The modular nature of our VR system enables games to be tailored for specific applications, such as training to use a virtual neuroprosthesis or brain machine interface (BMI) [8]. BMIs aim to accurately and reliably translate neural recordings into control signals for external effectors in order to replace lost sensory or motor function [9]. Controlling a closed-loop BMI via visual feedback is a learned skill that can be improved through training [10], though challenges remain. Achieving effective control of BMIs for real-world use can be time-consuming and requires the user to flexibly adapt to varied and unpredictable sensory conditions. Additionally, it is important that a sense of body ownership be extended to the effector, allowing it to be incorporated into the body schema [11]. Our VR system provides a platform where the visual environment and limb properties can be systematically altered to facilitate investigation of multisensory integration and the development of comprehensive training paradigms for BMIs. Here we describe the main features of this system and demonstrate its use in reaching experiments with NHPs.

II. METHODS

A. Animal Preparation

Two adult male NHPs (*Macaca mulatta*) participated in the study. The primates' welfare and all experimental procedures were carried out according to the U.S. Public Health Service Policy on Humane Care and Use of Laboratory Animals and the Guide for the Care and Use of Laboratory Animals and were approved by the Arizona State University Institutional Animal Care and Use Committee. Housing, feeding, and environmental social enrichment followed institutional standards accredited by AAALAC International.

B. Motion Capture

Kinematic data of the monkey's right arm movements were recorded using a reflective marker-based motion capture system (Motion Analysis Corp., Santa Rosa, CA). Included in the system are 10 *Kestral 2200* cameras capable of 300fps with a sensor resolution of 2048 x 1088 pixels. The system's interface was managed by Cortex *2016* (Motion Analysis Corp., Santa Rosa, CA), a motion capture editing software that controlled camera settings and recording/editing of captures. For tracking, five spherical reflective markers (12.5 mm in diameter) were sewn onto a custom-made sleeve fitted to the NHPs' right arms (Fig. 1a). Adhering the markers to a sleeve maintained consistent marker position for each

*Research supported by National Science Foundation Division of Integrative Organismal Systems Award 1558151.

K. Phataraphruk, P. VanGilder, and C. A. Buneo are with the School of Biological and Health Systems Engineering, Arizona State University, Tempe, AZ 85287 USA (phone: 480-727-0841; fax: 480-727-7624; e-mail: cbuneo@asu.edu).

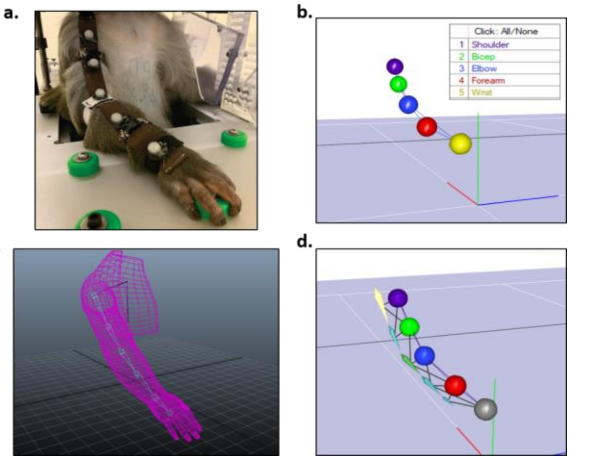

Figure 1. Process for driving avatar animation with kinematic data. (a) Motion capture sleeve attached to animal's right arm. This is a typical pose taken for the static capture. (b) Completed marker set with color coded markers connected by links. (c) Avatar model mesh. (d) Segment model contains bones attached (black lines) to the markers in the Model Pose. Bones pulled out for visualization but are normally embedded into the markers.

recording session. Markers were positioned on the shoulder, bicep, elbow, forearm, and wrist. Cameras were mounted in the front half of the animal testing room at various locations and oriented to create a 1150 x 700 x 300 mm (width, depth, height) capture volume surrounding the NHPs' workspace. Having multiple cameras at various locations ensured optimal marker detection for spatial precision; e.g. at least 3 cameras could continually detect each marker even if limb orientation during movement temporarily occluded one or more cameras.

Before recording, a "MarkerSet" was created in Cortex that identified the markers and their relative positions. The MarkerSet was constructed from a static capture recorded while a NHP sat at a table and held their arm at the starting position for approximately 2-3 seconds. A biomechanical model was then developed that labeled each marker and established links connecting them to each other (Fig. 1b). This capture also created the "Model Pose," used to create the segment model that renders the avatar arm. The MarkerSet was saved as a "Template" that allows the system to automatically identify the markers during real-time recording or capture.

To build a more robust template, the NHP's hand was then guided from the start position to each target for a series of 10-20 second dynamic captures, after which the animal was allowed to move their arm freely in the workspace to collect additional movement data. These dynamic captures were appended to the template and established the minimum and maximum ranges for the linkages between each marker during movement.

C. Avatars

Four realistic arm models were constructed by Baltu Studio LLC in Mesa, Arizona. Two NHP arms were designed to look almost identical to the individual NHPs using reference photos. A human male right arm was a composite design from multiple reference images. A prosthetic arm was modeled on the LUKE Arm (Mobius Bionics, Manchester, NH), a commercially available robotic upper-arm prosthesis.

D. Skeleton Model

A skeleton model allowed the avatar's animation to be driven by real-time kinematic data. Avatars were uploaded into Maya (Autodesk, Inc.) where the skeleton was constructed (Fig. 1c). Using the "Joint Tool," bones were positioned relative to each other in their approximate locations on the body. This tool automatically creates a "Joint Chain" that builds the "Skeleton Hierarchy" – parent and child joint relationships that define which joints drive translations and rotations of their corresponding dependent joints. Mesh and texture layers overlaid the skeleton model. The mesh layer forms the surface or "skin" of the avatar and creates its basic shape (Fig. 1c). The texture layer(s) contain features such as skin color, hair, and other realistic appearance details. The skeleton model was bound to the mesh layer of the avatar, enabling the joints to modify the appearance of the mesh (e.g. deformation) during animation. The skeleton model was exported from Maya as a Motion Analysis HTR file for use in Cortex (Cortex Plugin-in tool, Motion Analysis).

E. Segment Model and Animation

The "Segment Model" (Fig. 1d) used to drive the avatar's movements is created through the "Calcium Solver" embedded in Cortex. This tool utilizes marker data to calculate the skeleton's real-time motion. The Skeleton Model is aligned with the Model Pose and merged into the template. Each joint is attached to a corresponding marker that controls the movements of that particular joint. For better control, each joint is attached to two or three markers that are near that joint (Fig. 1d).

A joint type is selected for each joint that will determine its degrees of freedom. The Solver engine uses the joint parameters and marker data to calculate the kinematics of the skeleton model. During real-time motion capture, Cortex sends this kinematic data via SDK2 streaming to Unity3D, which renders the avatar within the VR environment.

F. VR Game

The VR environment, also created by Baltu Studios, was designed to resemble the testing room. Unity3D executed multiple C# scripts which communicated to other programs in our setup. A total of 4 scripts were utilized to control and monitor the activity in the game. "Cortex Connect" is a Motion Analysis Corp. script that contains the plugin which received the kinematic data streaming from Cortex via SDK2 and allows rendering of the avatars (Fig. 2a). "UDP Receive" creates a connection with LabVIEW 2017 (National Instruments, Inc.) to receive information regarding the task parameters (Fig. 2b). LabVIEW controlled the entire task including randomizing trials, difficulty level, onset and offset of targets, and monitoring of task performance. In conjunction, "UDP Send" sent the x,y,z position of the avatar's hand to LabVIEW to check that it reached the desired target (Fig. 2c). The last script is "Arm Settings," for quick selection of avatar arms and their visual properties (Fig. 2d). Currently, there are 4 avatars in the system, but additional avatars can be easily added and then selected via a drop-down menu. Regarding visual properties, blurring levels of 25%,

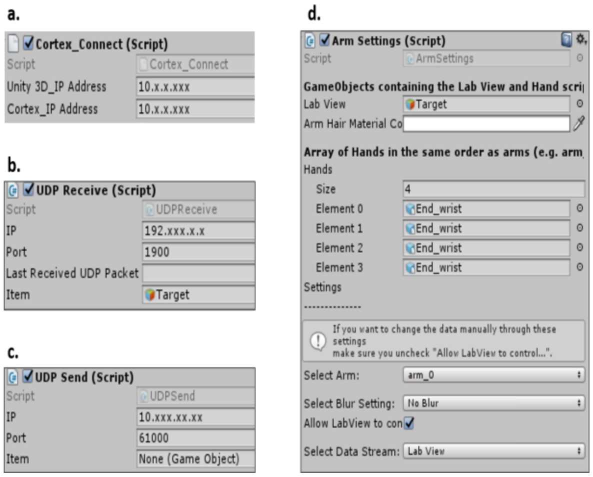

Figure 2. Unity3D game interface showing scripts that communicate to other programs in the setup. (a) Cortex Connect script specifying IP addresses for the workstations. (b) UDP Receive script specifying the connection protocols necessary for LabVIEW to control the specified Item (Target) in Unity3D. (c) UDP Send script specifying the connection protocol for communication from Unity3D to LabVIEW. The Item parameter is automatically updated to the hand of the selected arm. (d) Arm Settings script for selection of avatar arms and their visual properties. “Arm Hair Material Color” is a hex color code specifying the blurring effect. “Hands” is an array containing all the bones associated to the hand of each avatar.

50%, or 75% can currently be selected. This is useful for experiments investigating multisensory integration as blurring the arm alters its visual reliability.

G. System Architecture

The system utilizes two computers and a PXI embedded controller (National Instruments, Inc.) connected via ethernet cables to a switch box (NetGear 5-Port Gigabit Ethernet Unmanaged Switch) (Fig. 3). Rapid communication was facilitated by using UDP connections to send packets of information to and from the computers. The main computer (Workstation 1) ran Cortex and interfaced with the LabVIEW programs, which were executed in real-time on the PXI system. After completing an experiment, kinematic data was saved on the PXI system and later extracted for analysis. Unity3D was run on a second computer (WorkStation 2) which also interfaced directly with the VR display.

H. Behavior Task

NHPs were seated at a table in front of a display mirror oriented at an angle of approximately 45 degrees. The mirror blocked the animal’s vision of their arm and reflected the VR environment and avatar projected from a 3D monitor placed above the table. NHPs were trained to perform reaches from a single starting position to four VR targets located along the plane of the table. Reaches were made with and without concurrent tactile target feedback. Successful trials were rewarded with a water juice mixture administered through a tube extending from the mirror.

Tactile feedback was manipulated in blocks that alternated daily. For blocks with concurrent tactile feedback, round plastic objects approximately 34mm in diameter were placed on the table at the VR target positions. The VR targets were designed to look identical to the physical targets. To examine effects of visual reliability, the avatar arms were randomly

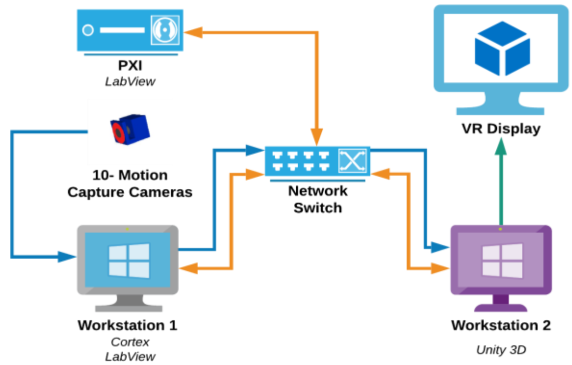

Figure 3. System architecture. Blue Arrows: motion capture data from cameras is sent to Workstation 1 for processing in Cortex and then outputs to Workstation 2 to drive the avatar arm in Unity3D. Orange Arrows: data sent and received from LabVIEW. Green Arrow: display data to 3D monitor.

rendered in either a clear vision state or maximum blurred visual state (75%) on a trial by trial basis.

III. Results

A. Avatar Rendering

Four avatars (including separate models for each NHP) were rendered in the VR with movements controlled by the NHP’s arm motions (Fig. 4a-c). The blurring feature manipulated the avatar’s texture with a “material”. Materials are features that can be added to a texture to alter the avatar’s appearance. This blurring material controlled by the “Arm Settings” script adds a blurring effect to the selected arm (Fig. 4d). This blur is maintained over the entire right arm of the avatar until LabVIEW sends a command to disable it.

B. Reaching Behavior

We used the system to conduct a preliminary investigation of the accuracy and precision of reaches made to VR targets under different sensory feedback conditions. NHPs made reaches with either no blurring or maximum blurring on the VR NHP avatar arm. Tactile feedback of reach targets was provided in alternating blocks of trials, and blurring levels were interleaved within blocks.

In the tactile condition, hand paths for monkey P were relatively consistent (Fig. 5a) but the removal of tactile feedback resulted in hand paths that were more variable (Fig. 5b). Reach endpoints also showed a similar pattern, with data points more tightly clustered in the tactile condition (Fig. 5c) compared to the non-tactile condition (Fig. 5d).

The distributions of reach endpoint positions were compared statistically between the tactile and non-tactile sensory conditions for each visual condition. To do this we first computed the minimum 3D ellipsoid volume that encompassed all the endpoints for a given target and set of sensory conditions. Ellipsoid volumes were then compared using permutation tests. Volumes were generally larger for the non-tactile conditions (Fig. 5d), though these differences were only statistically significant for select targets (Table 1). Effects of blurring seemed to be more apparent for Monkey P as indicated by the statistical significant differences in the blur condition for targets 1-3 (see also Fig. 5d).

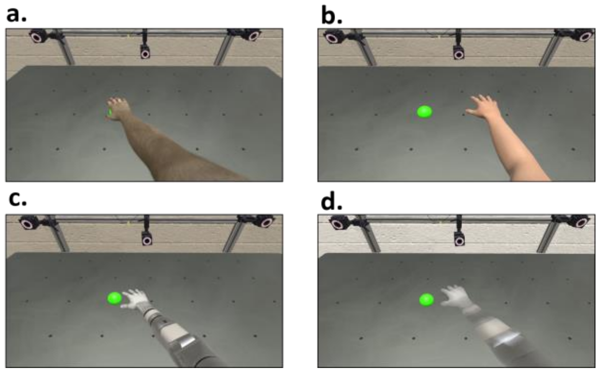

Figure 4. Avatar arms rendered in the VR environment. (a) Monkey Q avatar arm acquiring and holding at target 1 (b) Human avatar arm initiating reach to target 1 (c) LUKE Arm reaching to target 1. (d) Same as (c) but under max blurring.

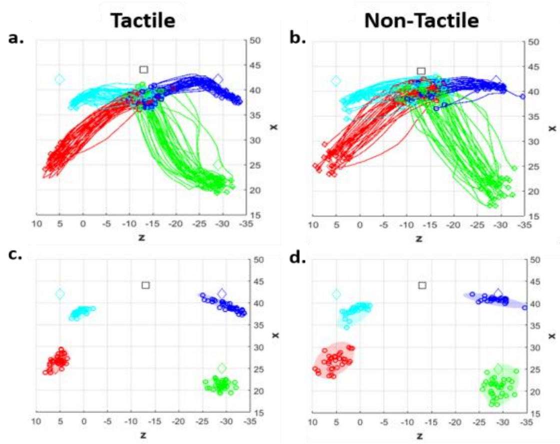

Figure 5. Example handpaths and reach endpoints from Monkey P in the max blur condition. Reaches were made from a start position (black square) to targets 1-4 (diamonds; numbering is clockwise from top left). Units are arbitrary units from VR coordinate system. (a) Handpaths from the tactile condition. (b) Non-tactile handpaths. (c) Reach endpoints and encompassing 2D ellipses for the tactile condition. (d) Non-tactile endpoints and ellipses.

Table 1		Target 1	Target 2	Target 3	Target 4
Monkey P	Vision	0.398	0.501	0.097	0.012
	Blur	0.0001	0.054	0.012	0.111
Monkey Q	Vison	0.386	0.512	0.001	0.111
	Blur	0.113	0.46	0.206	0.305

Table 1. P-values for permutation tests comparing tactile and no tactile conditions in the Vision and Blur condition. Grey boxes indicate statistically significant differences ($p<0.05$)

IV. Discussion

We have described and demonstrated a flexible VR platform that involves a small number of multiple interacting computers and programs and allows several visual parameters to be modulated with ease, making it useful for both motor control and perceptual studies. Its versatile design could also make it effective as a component in systems for closed-loop prosthetic training and rehabilitation. For this application, a method must be incorporated to drive the avatar arm using decoded neural signals rather than via recorded limb movements. Such methods already exist and have been described elsewhere [8].

Although many VR game engines exist, adopting Unity3D as our VR engine was a key component in making this setup possible. Its interface is intuitive and accepted coding languages include C#, JavaScript, and Boo (Python based). Unity3D also has available plugins for VR hardware such as the Oculus Rift (Oculus VR LLC., Irvine, CA) or can be exported to mobile devices or gaming platforms (i.e., Xbox). The versatility of this game engine makes it ideal for the design of games for rehabilitation, neuroscience and biomedical engineering applications.

Our preliminary data demonstrates that altering visual reliability and modulating tactile feedback does influence reaching performance. This suggest the system can be used to study the neural mechanisms of multisensory integration for the upper limb. However, results did differ between animals and visual feedback conditions and across targets. For example, blurring resulted in fairly consistent effects across targets for Monkey P but not for Monkey Q. This could reflect differing perceptual thresholds for blurring across animals; higher blurring levels might lead to more consistent effects and will be explored in future experiments. We will also experiment with blurring applied to different avatar arms (e.g. human or prosthetic) that are controlled by an animal's movements, in order to probe interactions between multisensory integration and body ownership.

Acknowledgment

The authors thank Bijan Pesaran (New York University) for his input on the system design, Emily Schaefer (Motion Analysis Corp.) for assistance with motion capture and Peter Costa, Dennis Bonilla, Laurie Annis, and Paul Diaz (Baltu Studios) for their work on the VR. The authors also thank Amanda Knight and Sydney Gowen for technical assistance.

References

[1] A. A. Faisal, L. P. J. Selen, and D. M. Wolpert, "Noise in the nervous system," *Nat. Rev. Neurosci.*, vol. 9, no. 4, pp. 292–303, 2008.

[2] M. O. Ernst and H. H. Bülthoff, "Merging the senses into a robust percept," *Trends Cogn. Sci.*, vol. 8, no. 4, pp. 162–169, 2004.

[3] D. C. Knill and A. Pouget, "The Bayesian brain: The role of uncertainty in neural coding and computation," *Trends Neurosci.*, vol. 27, no. 12, pp. 712–719, 2004.

[4] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, no. 6971, pp. 244–247, Jan. 2004.

[5] D. E. Angelaki, Y. Gu, and G. C. DeAngelis, "Multisensory integration: psychophysics, neurophysiology, and computation," *Curr. Opin. Neurobiol.*, vol. 19, no. 4, pp. 452–458, 2009.

[6] Y. Shi, G. A. Apker, and C. A. Buneo, "Multimodal representation of limb endpoint position in the posterior parietal cortex," *J. Neurophysiol.*, vol. 109, no. 8, pp. 2097–2107, 2013.

[7] P. VanGilder, Y. Shi, G. A. Apker, K. Dyson, and C. A. Buneo, "Multisensory interactions influence neuronal spike train dynamics in the posterior parietal cortex," *PLoS One*, vol. 11, no. 12, pp. 1–18, 2016.

[8] D. Putrino, Y. T. Wong, A. Weiss, and B. Pesaran, "A training platform for many-dimensional prosthetic devices using a virtual reality environment," *J. Neurosci. Methods*, vol. 244, pp. 68–77, 2015.

[9] A. B. Schwartz, "Cortical Neural Prosthetics," *Annu. Rev. Neurosci.*, vol. 27, no. 1, pp. 487–507, 2004.

[10] J. M. Carmena *et al.*, "Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates," *PLoS Biol.*, vol. 1, no. 2, p. e42, Oct. 2003.

[11] C. D. Murray, "An interpretative phenomenological analysis of the embodiment of artificial limbs," *Disabil. Rehabil.*, vol. 26, no. 16, pp. 963–973, Aug. 2004.