VRSurus: Enhancing Interactivity and Tangibility of Puppets in Virtual Reality

Figure 1: The top figure shows a puppeteer wearing an elephant puppet with VRSurus playing a VR game. The bottom figure shows the stereo rendering results in the virtual reality headset (Oculus Rift DK2).

Ruofei Du*
Augmentarium Lab | UMIACS
Computer Science Department
University of Maryland
College Park, MD, USA
ruofei@cs.umd.edu

Liang He*
Makeability Lab | HCIL
Computer Science Department
University of Maryland
College Park, MD, USA
lianghe@cs.umd.edu

* indicates equal contribution.

CHI'16 Extended Abstracts, May 7–12, 2016, San Jose, CA, USA.
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2892290

Abstract
We present VRSurus, a smart device designed to recognize the puppeteer's gestures and render tactile feedback to enhance the interactivity of physical puppets in virtual reality (VR). VRSurus is wireless, self-contained, and small enough to be mounted upon any physical puppets. Using machine-learning techniques, VRSurus is able to recognize three gestures: swiping, shaking and thrusting. Actuators (*e.g.,* solenoids, servos and vibration motors) assist with the puppetry visible to the audience and provide tactile feedback on the puppeteer's forearm. As a proof of concept, we implemented a tangible serious VR game using VRSurus that aimed at inspiring children to protect the environment and demonstrated it at the ACM UIST 2015 Student Innovation Contest. Our 3D models, circuitry and the source code are publicly available at www.vrsurus.com.

Author Keywords
Virtual Reality; Tangible User Interface; Haptics; Gesture Recognition; Head-Mounted Display

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation (e.g.,HCI)]: User Interfaces: Input devices and strategies; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism: Virtual reality

Figure 2: Closer view of an elephant puppet with VRSurus. The elephant puppet wears a custom 3D-printed cap that encapsulates the Arduino microcontroller, an inertial measurement unit and other electronic modules.

Introduction

Puppets are widely used in storytelling for puppetry on the stage as well as playing games among children. However, as physical inanimate objects, conventional puppets can hardly provide immersive auditory and haptic feedback. Previous research such as 3D Puppetry [6] and Video Puppetry [1] used a *Microsoft Kinect* or an overhead camera to create digital animation with rigid puppets. Nevertheless, they did not allow users to *wear* a flexible puppet in an immersive virtual reality environment. As exploratory work, we started with the following questions: “Could we endow puppets with more interactivity in both real and virtual world? Could we make traditional puppetry more immersive with virtual reality? Could we allow puppeteers to feel the *life* of puppets via haptic feedback?”

To answer these questions, we have designed and developed VRSurus (Figures 1 and 2), a smart device that enhances interactivity and tangibility of puppets using gesture recognition and tactile feedback in a virtual reality environment. VRSurus uses simple machine-learning algorithms and an accelerometer to recognize three gestures. It is also built with servo motors, solenoids and vibration motors to render haptic sensations. VRSurus is designed in the “hat” form so that it could be mounted upon any puppet. We have also implemented a serious VR game where the puppeteer manipulates a virtual elephant puppet (corresponding to the physical puppet) to clear the litter in the forests, splash water onto the lumbermen and destroy factories that pollute the air. The game is designed to educate children on environmental protection.

Our main contribution is the concept of an interactive puppet for virtual reality powered by tactile feedback and gesture recognition, as well as our specific mechanical and software design. The main benefit of our approach is its applicability to VR games and puppetry performance. We have presented this game at the *ACM UIST 2015 Student Innovation Contest* in Charlotte, North Carolina, USA. Please watch our supplementary video for a demonstration at www.vrsurus.com.

Related Work

VRSurus was built upon previous work in smart puppetry, gesture recognition, tangible interfaces and virtual reality.

Smart Puppetry

In this paper we use the term **smart puppetry** to refer to novel approaches that combine puppets and electronics. Our work is largely inspired by *3D Puppetry* [6] invented by Held *et al.*, which uses puppets as input via a *Microsoft Kinect* depth sensor to create virtual animations. Their smart puppetry allows a puppeteer to manipulate multiple rigid puppets simultaneously. However, it requires pre-creation of the puppet database with visual features such as SIFT. Similarly, *Video Puppetry* [1] developed by Barnes *et al.*, allows puppeteers to perform puppetry with paper cut-out characters and an overhead camera but also demands a pre-trained SIFT database. Unlike previous work, VRSurus does not require such visual databases. Thus, it can be easily adapted to other puppets without extra training efforts.

Our work is also related to an embodied smart puppetry called *3DTUI* [12], which was developed by Mazalek *et al.*. *3DTUI* allows users to pull strings above a cactus marionette to control a virtual character. Their follow-up work *I'm in the Game* [11], manipulated a virtual character using a puppeteer's full body movement by strapping a wooden puppet to the puppeteer's body. In VRSurus, the animation of the virtual character is controlled by both the head-mounted display and the inertial measurement unit (IMU) in

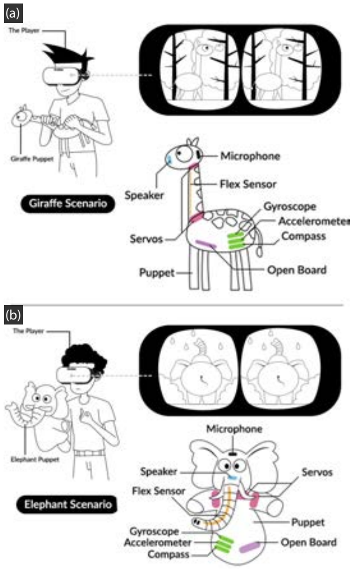

Figure 3: Sketches of our conception of VRSurus. (a) shows a puppeteer controlling a giraffe puppet to play a platform VR game; (b) shows a puppeteer controlling an elephant puppet to play a first-person VR game. Initially, we planned to sew sensors directly onto the puppet. In subsequent iterations, we created a hat-like attachment which allows VRSurus to be mounted onto any puppet.

VRSurus. We create a ride-on-puppet perspective by synchronizing the puppeteer's viewpoint and the orientation of the puppet in real time, which is deployed by the quaternion rotation from the head-mounted display (HMD). We use the IMU in VRSurus to recognize gestures and perform actions such as swiping, thrusting and shaking.

Previous research also investigates novel approaches that use data gloves [15], smartphones and craft supplies [10] or markers and robots [18] to perform virtual or physical puppetry. In contrast, our system preserves the natural user experience of **wearing** an off-the-shelf puppet. VRSurus serves as an attachment that enhances the interactivity of puppets instead of replacing it.

Tangible Interfaces for Virtual Reality
The tangible interface of VRSurus is mostly related to *Impacto* [9] invented by Lopes *et al.*. *Impacto* decomposes physical impact into a tactile aspect using a solenoid and an impulse response using electrical muscle stimulation. Similar to *Impacto*, we use two solenoids to tap the puppeteer's skin and use vibration motors for impulse feedback. To create physical animations, we use servos to actuate parts of the puppets (e.g., the elephant's ears). Similarly, tapping is also used by Li *et al.* to convey messages [8].

Besides *Impacto*, previous research on tangible interfaces for virtual reality also inspired us in our design process and future plans. For example, Kron and Schmidt [7] developed a haptic glove which provides heat, vibration and kinesthetic feedback via miniature tactile fingertip devices for virtual and remote environments. Bao *et al.* invented *REVEL* [2], which uses reverse electrovibration to create an oscillating electrical field around the user's fingers for augmented reality use cases. Schmidt *et al.* presented *Level-Ups* [17], a pair of interactive stilts which stimulates haptic feedback for walking up and down the steps in virtual reality.

Gesture Recognition for Virtual Reality
Gesture recognition has been well explored in the past couple of decades [13]. Previous work in gesture recognition usually involves external devices such as color or infrared cameras [14], depth sensors [16], data or conductive gloves [19, 20] or inertial sensors [3, 4]. For puppetry in the virtual reality environment, we employ low-cost, cheap, and reliable IMU, which can also be integrated within the puppet, to train our own classifier for gesture recognition.

System Design

We present the iterative design process of VRSurus along with the lessons learned throughout the procedure.

Ideation
In the initial ideation phase, we sketched two scenarios in Figure 3, the first "giraffe" scenario is similar to many platform games with a side-scroller effect such as *Super Mario World*. However, during the initial trials among members of our lab, we found that the 2D side-scroller design was not preferred because it did not fully utilize the 360 degree virtual world. Several lab members soon lost their interests because of the weak connection between themselves and the virtual character.

Therefore, we turned to the second "elephant" scenario, which is analogous to playing a first-person shooter game. When the puppeteer rotates his or her body, the virtual elephant moves together with the viewpoint as if the puppeteer is "riding" on the elephant, thus enhancing the connection between the puppeteer and the virtual character. VRSurus is designed to recognize three gestures, mapping to three different animations as shown in Figure 6.

Instead of sewing all the sensors inside the puppet, as we did at an earlier design stage, we fabricated a smart hat encapsulating all external sensors as shown in Figure 5,

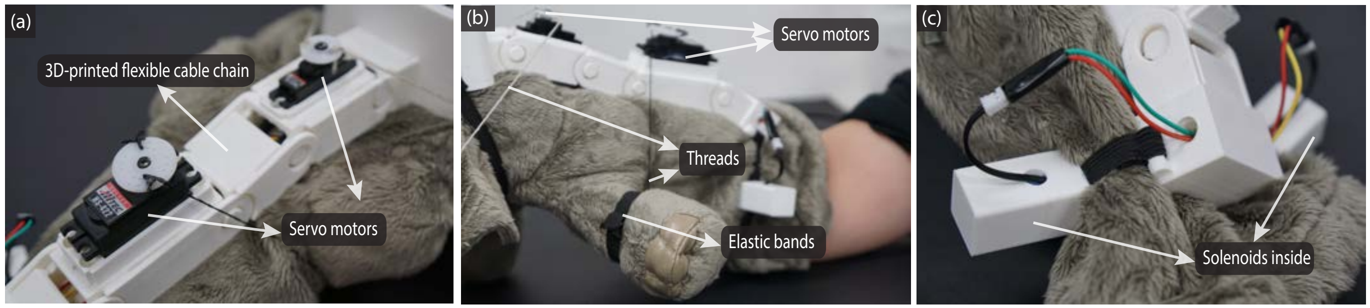

Figure 4: Mechanical design of VRSurus in details: (a) shows how two servo motors are seated inside the 3D-printed flexible cable chain, (b) shows how servo motors pull the forelegs of the puppet via threads and elastic bands, and (c) shows how two solenoids were embedded inside the 3D-printed cases around the puppeteer's forearms.

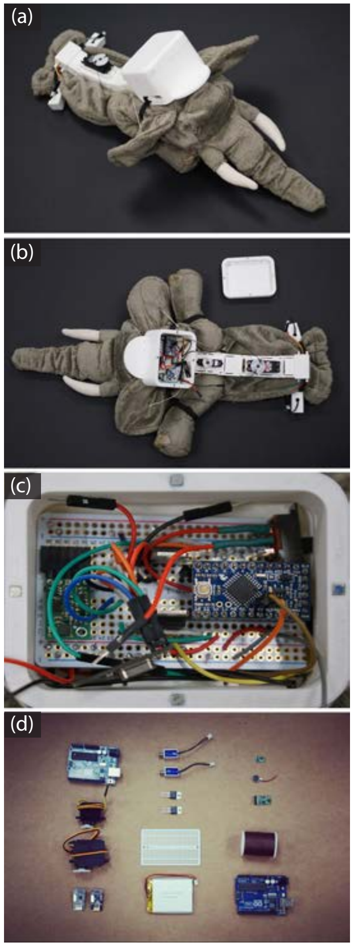

Figure 5: (a) shows VRSurus's profile view, (b) shows the orthogonal view with the cap removed, (c) shows the close-up view with the battery removed, and (d) shows each individual component.

which makes VRSurus usable in more puppets. We did not employ external sensors such as *Microsoft Kinect*, *Intel Real Sense*, and *Leap Motion Controller* to recognize gestures because they usually have a narrow field-of-view and limit the interaction space. We would like puppeteers to freely interact in the 360 degrees virtual reality.

Hardware Design
To empower the puppet's I/O abilities, we have leveraged Pololu MinIMU-9[1] (only accelerometer is used in the current prototype; gyroscope and compass are reserved for future use) to capture the puppeteer's gestures (input) and small push-pull solenoids, servo motors, and vibration motors to simulate tactile feedback and tangible animations on arm (output). Two solenoids, one on either side of the forearm, indicate the directions of the targets. Two servo motors may pull parts of the puppet using a string to simulate physical animations. Additionally, vibration motors can generate vibration feedback on the entire puppet.

Besides the basic I/O parts, VRSurus communicates with the software on a laptop via wireless connectivity. All electronic components are connected to an Arduino Pro mini. Two LiPo batteries are used to power the micro-controller and the wireless module individually.

Considering that all sensors and actuators should be mounted on the puppet, we have fabricated a flexible cable chain because (a) it can adapt to various sizes of the puppeteers' arms, and (b) the hollow structure of the cable chain unit is perfect for housing bigger components (e.g., motors). Both the hat-like case and the cable chain are 3D printed using PLA in an Ultimaker Printer[2]. All component specifications, circuitry schematics and 3D print files are available on https://github.com/ruofeidu/ninjaterp.

Software Design
The software part of VRSurus consists of a gesture recognizer, a proof-of-concept VR game and a web server.

[1] Pololu MinIMU-9: https://www.pololu.com/product/1265

[2] Ultimaker printer: https://ultimaker.com/

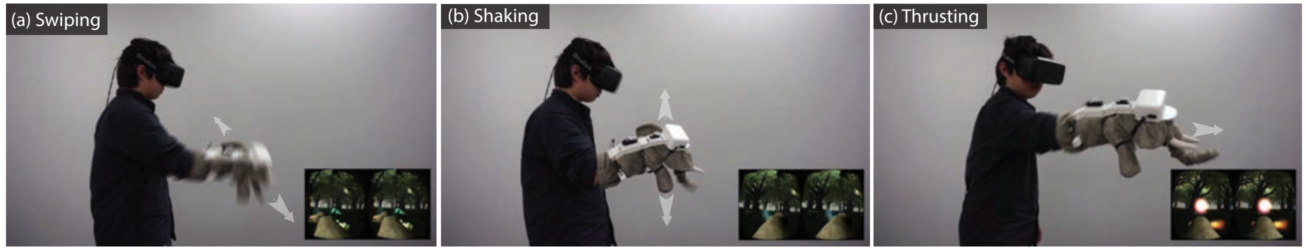

Figure 6: This figure shows three gestures we designed to control the virtual character: (a) swiping from left to right triggers the character to swing the nose and blow wind, (b) shaking up and down triggers the character to splash water, and (c) thrusting from inward to outward triggers the character to howl, trample on the ground and throw fireballs. Live demonstration is included in the supplementary video.

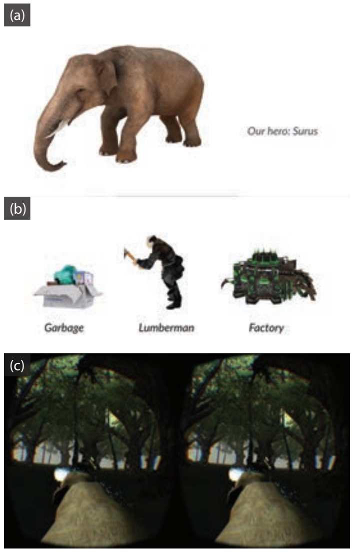

Figure 7: Virtual characters in the VR game. (a) shows the hero *Surus* controlled by the puppeteer, (b) shows the the enemies, (c) shows the Oculus Rift HMD rendering results of the lumberman being defeated by Surus.

Gesture Recognizer

We have trained our classifier using the decision tree algorithm in Weka [5]. In total, we used the following sixteen features: the sum of mean values on all axes, the differences between each each pair of axes, the power of each axis, the range of each axis, the cross product between each two axes and the standard deviation of values along each axis. To recognize the four gestures: idle, swiping, shaking and thrusting, we collected 240 sets of raw accelerometer values for each gesture from 4 lab members (60 sets per gesture per person). We achieved an average of 97% accuracy using 5-fold cross validation. The classifier and all signal processing program is written in Java.

VR Game

We designed and implemented a proof-of-concept serious VR game based on WebGL and WebVR written in Javascript and PHP. We use `Three.js`[3] for cross-platform rendering and `Sea3D Studio`[4] for rendering the animation in WebGL. In the VR game, the player acts as a little elephant called *Surus* to prevent evil human beings from invading the forest. The player is able to use defined gestures to commit attacks as stated in the Ideation section. The target can be litter, a lumberman and a polluting factory. The goal of the game is to defeat as many enemies as possible in a limited time. We designed a tutorial session before the game, encouraging players to familiarize themselves with each gesture and all the enemies. When a lumberman appears, the game would play the 3D audio of axe chopping according to the orientation between the player and the lumberman. Meanwhile, the game sends signals to VR-Surus to enable the left or right solenoid to tap on the user's arm. When the factory is destroyed, the game would output vibration feedback along with the 3D audio of an explosion. After each gesture is successfully performed, the servo motors are signaled to string the elephants' ears and forelegs backward and forward, simulating physical movements for the puppet. We have also written a web-server in PHP that exchanges signals between the gesture recognizer and the VR game every 10 ms. The game runs in 60 fps.

[3]Three.js: http://www.threejs.org
[4]Sea3D: http://sea3d.poonya.com

Figure 8: During the preliminary deployment, participants from the UIST community interacted with our prototype of VRSurus at the ACM UIST 2015 Student Innovation Contest

Preliminary Deployment

VRSurus was initially deployed for the ACM UIST 2015 Student Innovation Contest in Charlotte. It was demonstrated live throughout the entire 2.5-hour contest. Each game session lasted about two minutes (one exact minute for the VR game and about one minute for the introduction and hardware preparation). In total, there were 63 participants from the UIST community including two invited K12 specialists, who tried out our device. There were more than a hundred people who watched our device as well as the gaming procedure. Overall, the audience's reactions were positive. After successfully destroying an enemy in the VR environment, many participants cheered and some even laughed loudly. Some people mentioned that the tutorial greatly helped them learn the gestures and the virtual reality environment was really impressive. People were also greatly encouraged when the game told them that how many trees they "saved with their own hands (gesture)" in the end.

We also received some criticisms and suggestions for future work. One major concern was that the current prototype is a little fragile and heavy to wear over an extended period, although the device is small enough to be put upon the puppet. Some people also mentioned that the gesture recognizer failed to recognize their gesture when moving slowly and the tutorial did not give them any feedback. So during the deployment, we sometimes instructed people to move faster to enable the accelerometer to generate better data. In our next iteration, we plan to train the recognizer with more gestures from more volunteers and at varying rates of movements.

Conclusion and Future Work

In this paper, we have described a prototype device that enhances the interactivity of puppets with gesture recognition and haptic feedback for virtual reality gaming. We received initial feedback from participants at the ACM UIST 2015 Student Innovation Contest in Charlotte.

We have identified several directions for future work. One is to extend more input modules for better animations. For example, by embedding flex sensors in different parts of the puppet (e.g., ears and nose), the virtual character can animate corresponding parts when the puppeteer manipulates the physical one in the real world. By using a microphone, we can facilitate performance which kinetic input cannot support, such as creating a conversation with other virtual roles. Additionally, it could be a promising storytelling tool to express children's creativity. For instance, a child can experience others' stories with recorded sound, tactile feedback and physical movements all at the same time.

We also intend to conduct a long-term case study to examine how children would like to interact with smart puppets and what they can learn from immersive gaming experiences.

Acknowledgement

We would like to thank *Sai Yuan* from the Department of Animal and Avian Sciences in the University of Maryland for helping in sewing the puppets, as well as vocal and audio production in our demonstration software and video. We are also grateful to *Prof. Amitabh Varshney* for his suggestions that have helped improve the paper. We also thank the organizers of UIST Student Innovation Contest for support of the servo motors, as well as the UIST participants for precious feedback.

We appreciate the support of the NSF grant 14-29404 and the NVIDIA CUDA Center of Excellence. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the research sponsors.

REFERENCES

1. Connelly Barnes, David E Jacobs, Jason Sanders, Dan B Goldman, Szymon Rusinkiewicz, Adam Finkelstein, and Maneesh Agrawala. 2008. Video Puppetry: A Performative Interface for Cutout Animation. *ACM Transactions on Graphics (TOG)* 27, 5 (2008), 124:1–124:10.

2. Olivier Bau and Ivan Poupyrev. 2012. REVEL: Tactile Feedback Technology for Augmented Reality. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11.

3. Ari Y Benbasat and Joseph A Paradiso. 2002. An Inertial Measurement Framework for Gesture Recognition and Applications. In *Gesture and Sign Language in Human-Computer Interaction*. Springer, 9–20.

4. Keywon Chung, Michael Shilman, Chris Merrill, and Hiroshi Ishii. 2010. OnObject: Gestural Play With Tagged Everyday Objects. In *Adjunct Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 379–380.

5. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18.

6. Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D Puppetry: A Kinect-Based Interface for 3D Animation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software & Technology (UIST)*. ACM, 423–434.

7. Alexander Kron and Günther Schmidt. 2003. Multi-Fingered Tactile Feedback From Virtual and Remote Environments. In *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. 16–23.

8. Kevin A Li, Patrick Baudisch, William G Griswold, and James D Hollan. 2008. Tapping and Rubbing: Exploring New Dimensions of Tactile Feedback With Voice Coil Motors. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 181–190.

9. Pedro Lopes, Alexandra Ion, and Patrick Baudisch. 2015. Impacto: Simulating Physical Impact by Combining Tactile Stimulation with Electrical Muscle Stimulation. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST)*. ACM, 11–19.

10. Jesús Ibáñez Martínez. 2014. emoPuppet: Low-Cost Interactive Digital-Physical Puppets with Emotional Expression. In *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology*. ACM, 44:1–44:4.

11. Ali Mazalek, Sanjay Chandrasekharan, Michael Nitsche, Tim Welsh, Paul Clifton, Andrew Quitmeyer, Firaz Peer, Friedrich Kirschner, and Dilip Athreya. 2011. I'm in the Game: Embodied Puppet Interface Improves Avatar Control. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction (TEI)*. ACM, 129–136.

12. Ali Mazalek and Michael Nitsche. 2007. Tangible Interfaces for Real-time 3D Virtual Environments. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology*. ACM, 155–162.

13. Sushmita Mitra and Tinku Acharya. 2007. Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 37, 3 (2007), 311–324.

14. Kouichi Murakami and Hitomi Taguchi. 1991. Gesture Recognition Using Recurrent Neural Networks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 237–242.

15. Emil Polyak. 2012. Virtual Impersonation Using Interactive Glove Puppets. In *SIGGRAPH Asia 2012 Posters*. ACM, 31.

16. Zhou Ren, Junsong Yuan, and Zhengyou Zhang. 2011. Robust Hand Gesture Recognition Based on Finger-Earth Mover's Distance with a Commodity Depth Camera. In *Proceedings of the 19th ACM International Conference on Multimedia (MM)*. ACM, 1093–1096.

17. Dominik Schmidt, Robert Kovacs, Vikram Mehta, Udayan Umapathi, Sven Köhler, Lung-Pan Cheng, and Patrick Baudisch. 2015. Level-Ups: Motorized Stilts that Simulate Stair Steps in Virtual Reality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI)*. ACM, 2157–2160.

18. Ronit Slyper, Guy Hoffman, and Ariel Shamir. 2015. Mirror Puppeteering : Animating Toy Robots in Front of a Webcam. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI)*. 241–248.

19. John Weissmann and Ralf Salomon. 1999. Gesture Recognition for Virtual Reality Applications Using Data Gloves and Neural Networks. In *International Joint Conference on Neural Networks (IJCNN)*, Vol. 3. IEEE, 2043–2046.

20. Yupeng Zhang, Teng Han, Zhimin Ren, Nobuyuki Umetani, Xin Tong, Yang Liu, Takaaki Shiratori, and Xiang Cao. 2013. BodyAvatar: Creating Freeform 3D Avatars Using First-person Body Gestures. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 387–396.