AT LEAST, BE HUMAN: HUMANIZING THE ROBOT AS A MEDIUM FOR COMMUNICATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Michael John Pascual Suguitan

May 2022

AT LEAST, BE HUMAN:
HUMANIZING THE ROBOT AS A
MEDIUM FOR COMMUNICATION

Michael John Pascual Suguitan, Ph.D.
Cornell University 2022

I present an approach and case studies for humanizing robots through accessible design. Humanizing social robots is a central goal of human-robot interaction research and often involves a combination of two objectives: humanizing the mind through human-like intelligence, or humanizing the body through lifelike humanoid features. However, these literal approaches to humanizing pose technological and social challenges that have prevented adoption of robots in everyday social contexts. Even if convincingly humanized robots could be achieved, human-robot interaction may risk diminishing our capacity for human-human interaction. I propose to avoid these pitfalls by humanizing the robot as a medium for communication through accessibility. Accessibility can humanize technologies by makings their inner workings visible and familiar to human users, promoting understanding of the technological processes and imperfections. Accessibility also enables broader demographics of lay users to become involved with robotics, enabling communication *through* robots, from development processes to applications. I use the open-source Blossom social robot as an extended case study of this approach, and detail its technical implementations and research deployments. The goal of this work is to present accessibility as a way to humanize robots while enabling robot-mediated communication for human-human interaction.

BIOGRAPHICAL SKETCH

Michael is a roboticist from Manila, Philippines. He immigrated to Raleigh, North Carolina as a child, where he studied mechanical engineering and computer programming at North Carolina State University. Throughout his studies, he worked at NASA Marshall Space Flight Center, Samsung Research Center, Honda Research Institute Japan, and Facebook AI Research. His research interest is in combining art and technology in the design of intelligent robot companions, and enabling a broad community to participate in robotics.

He owes his interest in robots to a lifelong obsession with Japanese mecha anime, specifically *Voltes V*, *Gundam*, and *Evangelion*.

「本来は完璧なはずなのに、

どこかが壊れているとか僕は面白いと思う。

面白さってそう言う物だと思う。」

庵野秀明

ACKNOWLEDGEMENTS

"Words[1] *and pictures don't – they're like two different animals.*

They don't particularly like each other."

William Eggleston

[1] I would like to use words to express my gratitude to the many research assistants who made this work possible: Ankita Chandak, Harrison Chang, Cheng "CC" Chang, Emma Cohn, Rebecca Cooper, Greg Holman, Michael Hu, Miranda Jeffries, Antoine Klopocki, Preston Rozwood, Carolyn Shi, Ryan Tine, Dongqing Wang, and Angela Zhang.

Welc
to Da

Honda
'19 06 03

拝

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Part I

Introduction

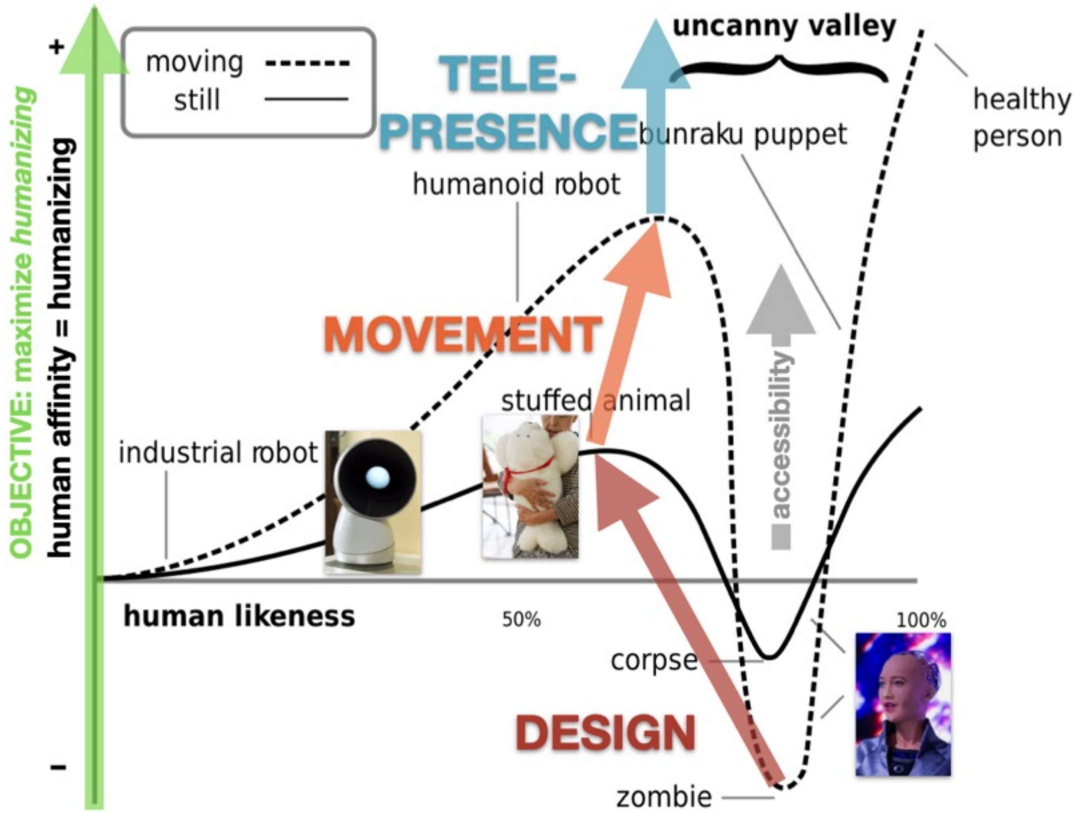


Figure 1: My proposed approach for humanizing the robot, referencing Mori's *bukimi no tani* (不気味の谷, "the valley of eerieness," anglicized as "the uncanny valley") as a conceptual framework [1]. I equate "humanizing" with maximizing "human affinity" on the vertical axis of the graph (left green arrow), and journey out of the valley by making accessible (gray arrow) three phases of robot development: design, movement, and telepresence.

0.1 Humanizing the Robot

In this section, I explain my interpretation of how to humanize robots through accessibility. I define the terms "robot," specifically in the context of social robotics, and "humanizing," which draws upon Masahiro Mori's concept of "the uncanny valley." I review current approaches and motivations for humanizing robots, which typically involve literal humanization through humanoid behaviors or appearances. I argue that, given the technological and social challenges of this literal approach to humanizing, we can instead humanize robots through alternative designs, specifically zoomorphism and accessibility. Zoomorphic designs have a humanizing effect by relaxing the expectations for interaction. Making phases of robot development accessible further humanizes by making the technology, including its processes and imperfections, familiar to lay users.

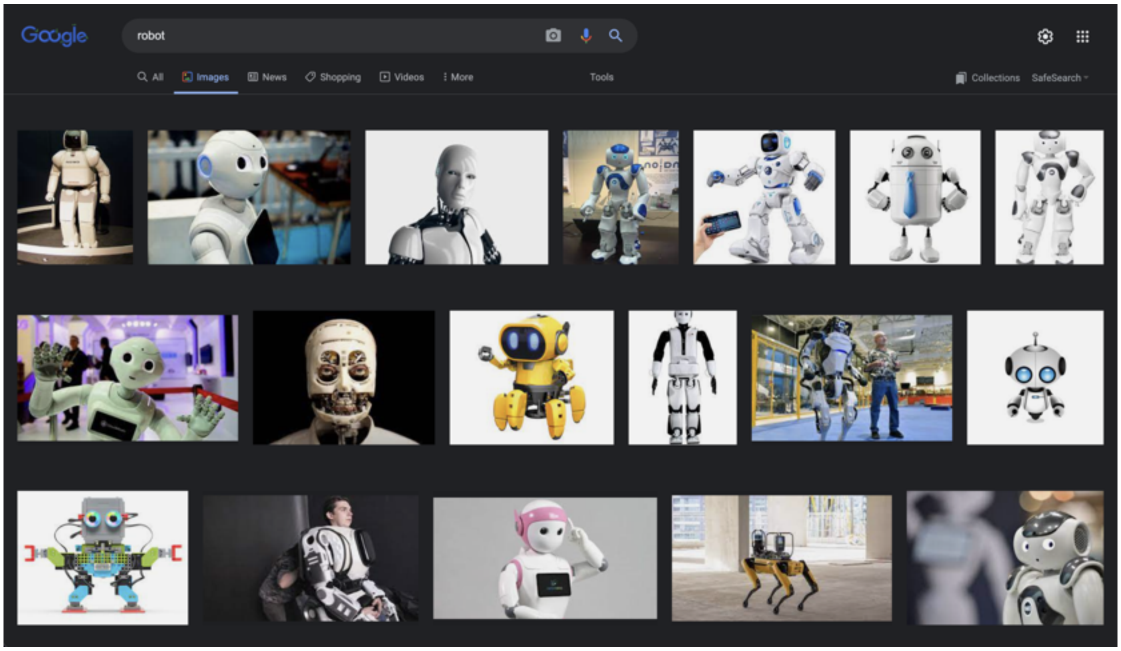
“Robot” is defined as “a machine that resembles a living creature in being capable of moving independently and performing complex actions; a device that automatically performs complicated, often repetitive tasks (as in an industrial assembly line)” [5]. Though the term carries etymological connotations of slavery and drudgery from its Czech roots [6], the field of “social robotics” orients away from utilitarian machines towards robots that perform social roles. The definition of “social robot” has been subject to several interpretations within human-robot interaction (HRI) research. Cynthia Breazeal, one of the pioneers of the field, defines a “sociable robot” as “socially intelligent in a human-like way, and interacting with it is like interacting with another person” [7]. Brian Duffy defines the social robot as “a physical entity embodied in a complex, dynamic, and social environment sufficiently empowered to behave in a manner conducive to its own goals and those of its community” [8]. Kerstin Dautenhahn and Aude Billard define “socially intelligent agents” as those which “show elements of human-style social interaction and behavior” [9]. Terrence Fong et al. define “socially interactive robots” as “robots for which social interaction plays a key role” [10], requiring that the robot exhibit “human social characteristics such as the expression and perception of emotions, high-level communication abilities, and a distinct personality that changes over time.” The prevailing notion common to all of these definitions is that the interactive capabilities of social robots should be humanized to some degree.

“Humanize” is defined as “to represent as human; to attribute human qualities to; to adapt to human nature or use” [11]. The term does not explicitly apply to only humans; we often humanize animals and objects by attributing human-like qualities to them. In the context of the prior definitions of “social robot,” humanization most readily translates to imbuing robots with the human qualities of humanoid appearances and human-like behavior. Jean-Christophe Giger et al. discussed humanization of robots from a psychological perspective, referring to “the effort to make robots that more closely mimic human appearance and behavior, including the display of humanlike cognitive and emotional states” [12]. Kate Keener Mays broadens the definition of “humanizing” to encompass sociological implications, such as the notion of robots’ social rights, and what our use of robots as technology reveals about our own human tendencies [13]. These definitions emphasize that humanizing is relative, dependent on both the design of the robot and the user’s expectations and interpretations.

In establishing the definition of “humanizing” that I use throughout this thesis, I reference Masahiro Mori’s concept of *bukimi no tani genshou* (不気味の谷現象, “the phenomenon of the valley of eeriness,” often anglicized as “the uncanny valley”) (Figure 1) [1, 14]. The graph’s horizontal and vertical axes, originally *ruijido* (類似同, “degree of similarity”) and *shinwakan* (親和感, “fellowship feeling”), are often anglicized as “human likeness” and “affinity,” respectively. “Affinity” itself is defined as “a liking for or an attraction to something; a quality that makes people or things suited to each other” [15]; this notion is similar to familiarity[2]. I argue that a “human” before “affinity” was lost in translation; reintroducing it yields “human affinity,” which I equate to “humanizing.”

[2] The first character of *shinwakan*, 親, can be read as “familiarity.” Karl MacDorman interpreted *shinwakan* as “familiarity” in his initial 2005 translation [16], then as “affinity” in his updated 2012 translation [17].

Thus, to humanize is to increase the feeling of human affinity and familiarity. I interpret humanizing as maximizing the vertical position on the uncanny valley (Figure 1, green arrow). As with Giger et al.'s definition, approaches to humanizing robots often comprise of two goals: humanizing the mind through familiar human-like intelligence, or humanizing the body through familiar lifelike humanoid appearance.



robot

Figure 2: The first Google image results for "robot," revealing a stark aesthetic uniformity: rigid plastic or metal light-colored bodies with illuminated accents.

0.1.1 Humanizing the Mind

Given the contemporary novelty of actual physical robots, science fiction has informed most public perceptions and expectations of robots [18]. A cursory web image search shows that contemporary perceptions of robots conform to aesthetics inspired by fiction (e.g. HAL and the EVA pod from *2001* [19], EVE from *WALL-E* [20]): rigid white bodies contrasted by black cutouts and illuminated

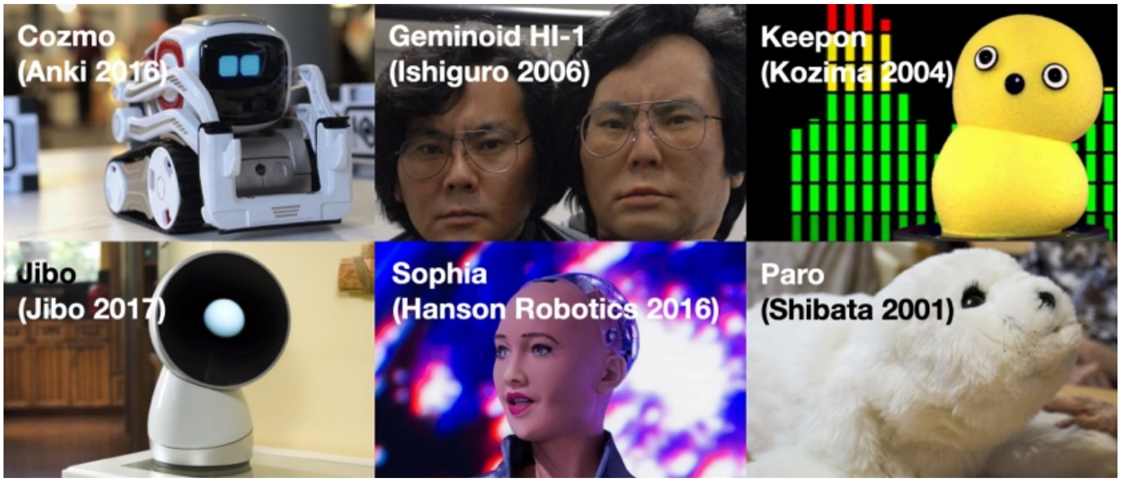


Figure 3: "Robotic" (Cozmo, Jibo) (left), humanoid (Geminoid, Sophia) (middle), and zoomorphic (Keepon, Paro) (right) robots.

accents (Figures 2 and 3, left). Though some of these robots are slightly anthropomorphized in their embodiments, they are largely humanized through their human-like behaviors by speaking and emoting in ways that are familiar to humans. The fictional inspirations are present in several consumer robots, such as Anki's Cozmo and Vector [21], Jibo [22], and Kuri [23]. Though these robots are technically sophisticated and well executed, the high expectations of their intelligence imprinted by fiction are nigh impossible to meet, and they have yet to live up to their promise as ubiquitous home companions [24, 25, 26].

0.1.2 Humanizing the Body

Inspired by other fictional works that critically question the boundary between natural and artificial humans [6, 27, 28, 29], humanoid robots such as the Geminoids [30] and Hanson Robotics' Sophia [31] are exemplars of humanizing robots through humanoid appearances (Figure 3, middle). Though technically impressive and lifelike, the humanoid aesthetic raises expectations for human users and interactors [32]. This same sophistication lends them to be mostly manually operated with few autonomous and actuation capabilities, limiting their capacity as independent social agents [33]. Similarly, as theorized by Mori, their imperfect approximation of human likeness in movement or appearance may yield “eeriness” that lands them in the uncanny valley [1].

0.1.3 Alternative Approaches to Humanizing

Humanizing the robot can be formulated as two separate Gordian Knots[3]: humanizing behavior tackles the knot of the artificial mind; humanizing appearance tackles the knot of the artificial body. Disappointment can arise from the gap between expectation and reality [34]; given that these approaches to humanizing robots are both technologically and socially intractable, I propose to instead cut both knots through zoomorphic robots (Figure 3, right). As an example from fiction, in *A.I.: Artifical Intelligence*, the zoomorphic Teddy is arguably the most human character in the film, serving as a loyal companion for the more humanoid but less humanized robot child, David. Though it seems counter-intuitive to humanize the robot through non-humanoid zoomorphism, we often humanize animals and pets. Kate Darling likens human-robot interaction to human-animal relationships with the potential for deep companionship [35]. Paradoxically, reducing the human likeness relaxes the expectations for interaction, potentially increasing the human affinity and familiarity (Figure 1, “stuffed animal” region preceding the fall into the valley). I place “robotic” robots on the initial gradual slope, zoomorphic robots on the local optima, and humanoid robots in the nadir of the valley.

[3] An intractable problem of untangling a complexly tied knot; the trick solution is often to just cut the knot.

Returning to the prior definition of “humanizing” as the effort to increase “human affinity or familiarity,” I argue that humanizing can be bolstered through accessibility. I take “accessibility,” defined as “capable of being used or seen; capable of being understood or appreciated” [36], to mean creating a robot platform that is understandable and familiar to non-roboticists. Whereas literal humanization (i.e. human-like behaviors and appearances) is closely aligned with anthropomorphism and its psychological mechanisms [37], accessibility humanizes by making technology familiar to lay users. This notion of humanizing technology through accessibility is encapsulated in the aesthetic of post-digital media. Music producer Kim Cascone coined the term “post-digital” in response to musicians developing an “aesthetic of failure:” treating glitch and imperfection from digital music production technologies as creative elements rather than unwanted artifacts [38]. The artifacts can originate from either the limits of the technology itself or from the technology’s accessibility to amateur users. The imperfections reflect the imperfection of the technology’s human creators and users. While the prefixing “post” connotes a return to “analog” processes, Florian Cramer argues that post-digital media *combines* analog and digital mediums to serve higher aesthetic goals while emphasizing processes over products [39]. Mattia Thibault offers Nintendo’s Labo construction kits as exemplars of post-digital aesthetics [40]: users construct do-it-yourself (DIY) cardboard housings for the gaming console, combining analog craft materials with digital electronic hardware to create physical interfaces, ranging from toy pianos to robot exoskeletons. Vygandas Šimbelis presents several post-digital works in his thesis titled *Humanizing Technology Through Post-Digital Art* [41]. Examples include *Metaphone*, a painting machine that uses biological sensors as input devices, and *STRATIC*, an audiovisual experience that digitizes analog light displays and audio sources. By emphasizing human inputs and making the system's inner workings visible, Šimbelis humanizes his works by "making digital technologies expose their imperfections and making them fathomable to human beings." The post-digital aesthetic provides a template for humanizing robots by making their processes familiar and embracing imperfections – those of the human creators and users and of the technology itself. The post-digital emphasis on technological mediums invites an interpretation of robots as mediums themselves.

0.2 Medium for Communication

In this section, I explain my interpretation of robots as mediums of communication. I first define the terms "medium" and "communication," drawing from media theory and the function of mediums in enabling communication. I then relate these theories to robots as mediums themselves, and present other arguments for their potential negative effects on communication. Finally, I reiterate that a potential way to circumvent these negative effects is by humanizing robots through accessibility.

"The medium is the message."

Marshall McLuhan [42]

"Medium" is defined as "a means of effecting or conveying something; a particular form or system of communication" [43]. Mediums can be any technology, artificial or natural, that enable communication, such as telephones which enable synchronous remote auditory communication. This document is

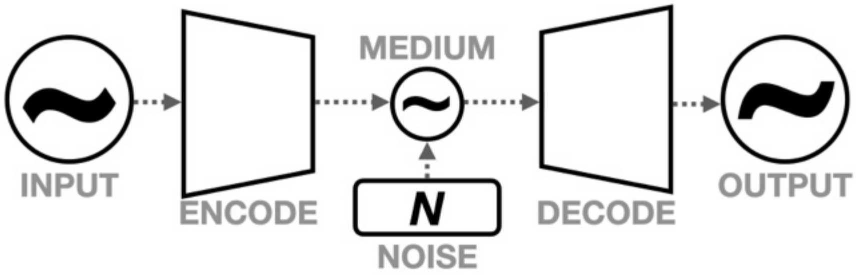


Figure 4: Shannon's mathematical model of communication.

a medium for the written word, which itself is a medium through which the authors communicate to readers, in a different space and at a different time. "Communication" itself is defined as "a process by which information is exchanged between individuals through a common system of symbols, signs, or behavior" [44]. I formalize "communication" through Claude Shannon's mathematical model of communication, wherein a sender sends a message that is encoded into a smaller embedded representation and decoded to the receiver on the other end (Figure 4) [2]. The message is the information passing through the medium, and the medium comprises of a compressed representation of the message subject to an additional noise source [4]. Marshall McLuhan, in claiming that "the medium is the message," argued that the medium is more important than any message passing through it [42]. McLuhan offers the simple lightbulb as an example: though potentially lacking any explicit message, it "creates an environment by its mere presence." The ability for the lightbulb to *define* the environment and terms of communication, McLuhan argues, is much more important than any message that a light source may convey (e.g. neon signs, illuminated billboards). A robot is also a medium that explicitly communicates its internal states through behaviors, while implicitly communicating its creators' notions of "robot" through its design.

[4] Alternative models of communication maintain the medium-message distinction [45].

Several scholars have written about robots as technological mediums. Darling, on the subject of zoomorphic robots used for healthcare, suggests that robots can “serve as a mediator, a conduit, of human-human interaction” [35]. Julia Hildebrand argues that we must pay attention to the ecological influences that robots have on ourselves and our way of being and communicating [46]. Sakari Taipale and Leopoldina Fortunati frame robots as a new class of information and communication devices, similar to mobile phones, with the potential to become the next “new media” [47]. Johan Hoorn has applied existing theories of computer-mediated communication in the context of robots as two distinct modes: robot-mediated communication (human users interact with other human users *through* the robot) and human-robot communication (human users interact *with* the robot as an autonomous agent) [48]. In this work, I largely focus on robot-mediated communication, partly due to the aforementioned difficulty in creating convincing human-like agents, but also due to the potential of human-robot communication to negatively affect our capacity for human-human communication.

> *“The nature of the medium...*
> *the amputation and extension of [their]*
> *own being in a new technical form.”*
>
> Marshall McLuhan [42]

McLuhan warns that though mediums can extend our communication abilities, they may also amputate it. For examples, the lightbulb extends our night sight but amputates our natural ability to see in the dark. In a modern setting, smartphones extend our ability to communicate across distances, but amputate our ability to converse face-to-face [49]. McLuhan warns that the benefits of extension through mediums "numbs" us, preventing us from noticing their capacity for amputation.

"There is a danger that the robots,
if at all successful, will replace people."

Sherry Turkle [50]

As with all mediums, robots pose risks for amputation. Sherry Turkle's ethnography of human-robot interaction illustrates the robot's capacity for social amputation [50]. Turkle recounts users wanting to replace their partners with sophisticated robots that always say "yes," becoming dependent on robots for matters as personal as health, or even creating robots as replacements for deceased family and friends. Cherie Lacey and Catherine Caudwell warn that robots, underneath their disarming and appealing aesthetics, may also conceal insidious intentions such as data logging or behavior shaping [51]. Other scholars have argued that offloading deeply human interactions to robots, such as therapy and caretaking, may be dehumanizing and diminishing for our capacity to interact face-to-face [52, 53, 54]. Robots as a medium may enable escapism for users to run away from complex and mutually hurtful human relationships towards the refuge of robots that neither tire nor disagree. HRI research, if successful in its goal of humanizing robotic companions through human-like behaviors and appearances, may yield a new medium that amputates our social capacity for human-human interaction.

I suggest that this amputation may be remedied by accessibility. Hildebrand suggests that we can avoid the numbing amputation of technology by "making media workings visible to us," echoing the post-digital notions of humanizing technology through familiarity and accessibility. Making the workings of technology accessible invites a broader audience to participate in its development, enabling communication between users through the technology as a medium. In the context of robots, I propose to reframe phases of robot development as forms of robot-mediated communication, including its physical construction, behavior authoring, and use as an embodied communication device.

0.3 Humanizing the Robot as a Medium for Communication

I formulate this goal as the overarching research question of this work:

How do we create and humanize robots as
mediums to extend human communication?

I use the Blossom robot, an open-source social robot platform I developed as a student, as an extended case study (Figure 5). I divide the approach to humanizing Blossom into three phases of its development – design, movement, and telepresence (Figure 5, left) – and detail the subsystems and interaction scenarios. Each phase emphasizes humanizing – "human affinity or familiarity" – by focusing on familiarity through accessibility. Blossom's design is accessible hardware that is open-source and user-customizable, consisting of a tensile interior actuation mechanism and soft exterior covers to achieve smooth, lifelike movements. Blossom's movement system is an accessible phone-based motion authoring system with back end behavior generation algorithms that expand upon the human-crafted behaviors. Blossom's telepresence capability is an ac-

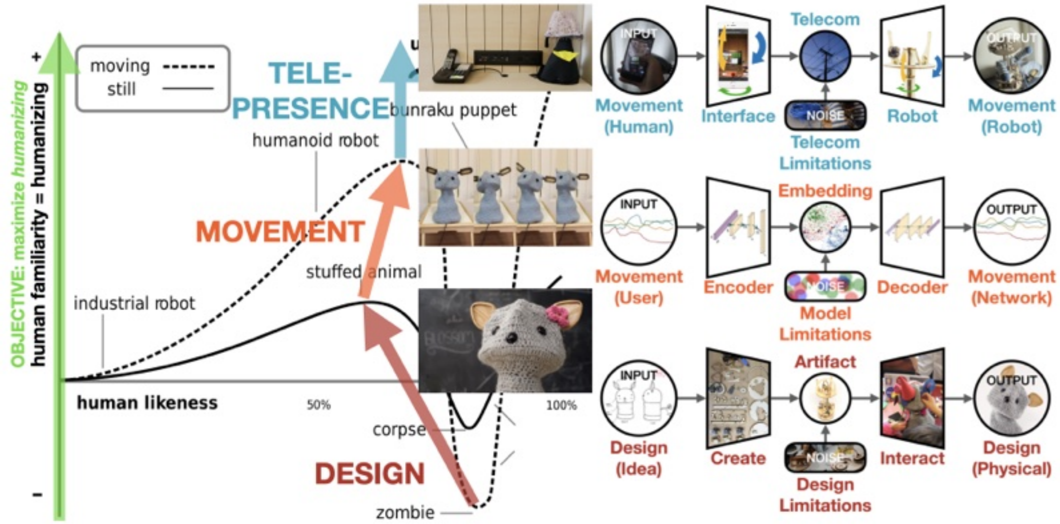


Figure 5: Blossom's journey out of the uncanny valley (left) and interpreting each phase (design, movement, telepresence) as forms of robot-mediated communication, as formalized through Shannon's model of communication (right) [2].

cessible front end motion-based teleoperation system that employs the remote user's proprioception. Involving users in each phase makes the inner workings of the system visible and familiar to lay users, achieving humanizing through accessibility.

Each phase further humanizes by framing the robot as a medium for communication between human users (Figure 5). Accessibility enables lay users to communicate *through* the robot in each phase. Blossom's design communicates user-defined perceptions of robot aesthetics, conveying diverse interpretations of what the robot could look like. Blossom's movement communicates user-crafted behaviors, conveying how users interpret how the robot should show affective responses. Blossom's telepresence capability communicates between users in remote locations, conveying human physicality at a distance. I apply the communication model throughout to frame each phase as a form of robot-mediated communication that extends human-human interaction.

The contributions of this work include:

- The design of Blossom, including physical artifacts and resources for reproducing the platform.
- A method for robot movement authoring using a phone-based authoring system and neural network-based behavior generation techniques.
- Descriptions of various deployments of Blossom, including case studies and user evaluations.

The overarching contribution is the case study of Blossom's development phases interpreted as forms of robot-mediated communication. The following sections of this thesis detail those three phases of Blossom's development (design, movement, and telepresence).

0.4 Design

"The research through design artifact is
the medium of generalizable knowledge."

Jeffrey Bardzell [55]

In this section, I provide an overview Blossom's design[5], including the aesthetic concepts that served as inspirations. I describe deployments of Blossom

[5] My contributions include: overseeing iteration through successive prototypes, from the initial 3D-printed prototype (designed by Greg Holman) to the laser-cut version (work carried out by Harrison Chang, Miranda Jeffries, and Rebecca Cooper) and successive refining; designing the electrical systems; creating the software library to control the robot (assisted by Michael Hu); designing workshops and demonstrations; providing support for other Blossom users.

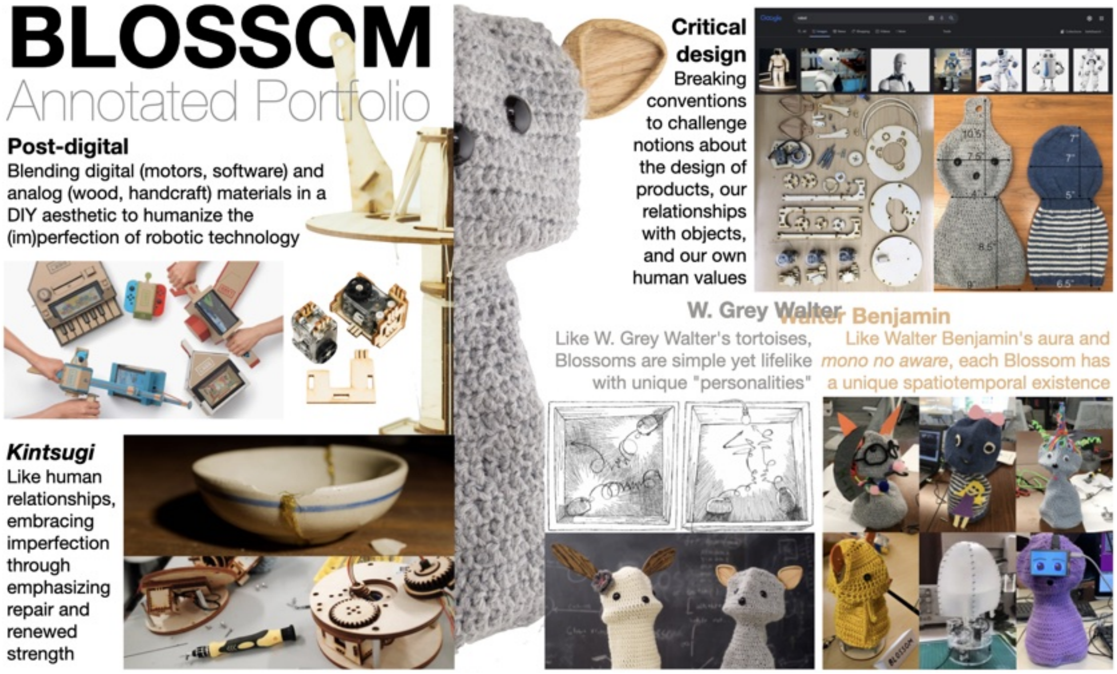

BLOSSOM
Annotated Portfolio
Post-digital
Blending digital (motors, software) and analog (wood, handcraft) materials in a DIY aesthetic to humanize the (im)perfection of robotic technology
Kintsugi
Like human relationships, embracing imperfection through emphasizing repair and renewed strength
Critical design
Breaking conventions to challenge notions about the design of products, our relationships with objects, and our own human values
W. Grey Walter
Like W. Grey Walter's tortoises, Blossoms are simple yet lifelike with unique "personalities"
Walter Benjamin
Like Walter Benjamin's aura and mono no aware, each Blossom has a unique spatiotemporal existence

Figure 6: Annotated portfolio [3] of Blossom juxtaposed against the aesthetic concepts that inspired its design.

in research and outreach contexts. I then frame Blossom's design as a form of robot-mediated communication: the design idea is transmitted through the medium of the physical artifact into a realization of the design.

0.4.1 Related

Research through design is a research paradigm that emphasizes the epistemological function of artifacts and focuses on "making the right things": products that transform the world into preferred states [56, 57]. Jeffrey Bardzell argues that the research through design artifact is itself the medium of generalizable knowledge [55]; knowledge is embodied in the artifact through its creation, and extracted by users through its consumption. Knowledge passes through the artifact, similar to how a message passes through a medium in the model of communication. Blossom's design as a physical artifact embodies several aesthetic concepts, which I will review in this section.

Post-digital aesthetics and *kintsugi*

As explained in the preceding sections, post-digital aesthetics serves as a reference for humanizing by making technology familiar and exposing its imperfections [38, 58, 39]. Similar to post-digital, *kintsugi* (金継ぎ, "golden repair") is a Japanese aesthetic concept that embraces imperfection by emphasizing repair and renewed strength [59]. These aesthetics encourage subjectively reinterpreting perceived shortcomings as celebrations of the artifact's history and relationships. Blossom's design exhibits post-digital traits by blending digital (e.g. motors, cables, software) and analog (e.g. wood, handcraft, strings) materials in a DIY aesthetic and user-involved process to humanize its design. Within social robotics, the OPSORO (OPen-SOurce RObot) social robot platform also served as an inspiration [60]. OPSORO's interior is comprised of reconfigurable modules, which inspired the snap fits and minimal hardware of Blossom's own interior mechanisms. Like *kintsugi*, Blossom's repairability lends the artifact a history and promotes ethical consumption and reuse of technology.

Critical design

Anthony Dunne and Fiona Raby define critical design as "speculative design proposals to challenge narrow assumptions, preconceptions and givens about the role products play in everyday life" [61]. Blossom's design is critical by its uniquely "non-robotic" appearance, looking less like contemporary consumer and fictional robots and looking more like a stuffed animal brought to life through a mix of traditional aesthetics and robotic technology. This unconventionality prompts users to critically question the stereotypical conformity of robot designs (Figure 2).

Robotic aura, or A Tale of Two Walters

W. Grey Walter invented the first “modern” robots: tortoises that simply moved towards or away from light sources [62]. Though simple, Walter remarked that the tortoises’ behaviors were convincingly lifelike, and further anthropomorphized them with names: “Elmer” and “Elsie.” Like Walter’s tortoises, Blossom is a simple assembly of motors and mechanisms, yet the result is more lifelike than its individual components belie.

In Walter Benjamin’s seminal 1935 essay “The Work of Art in the Age of Mechanical Reproduction,” he reflects on the then-burgeoning industrialization and its devaluing effects on an object’s aura, “its presence in time and space, its unique existence at the place where it happens to be” [63]. Benjamin’s aura is reflected in *mono no aware* (物の憐れ, “the pity of things”), another Japanese aesthetic trait that celebrates the ephemeral nature of existence. Like Benjamin’s aura and *mono no aware*, though Blossom’s design is infinitely reproducible, the marks of creation are present in the imperfections of its hardware and customization of its aesthetics. Accessibility and aura are balanced by providing the “template” for Blossom’s internal structure while keeping its exterior aesthetics open-ended.

We have deployed Blossom in workshops for middle schoolers to build and customize their own robots (Figure 6, bottom right, top row), and other research groups have reproduced Blossoms for their own research goals (Figure 6, bottom right, bottom row). Friedman et al. have used Blossom as a "canvas" to explore clothing for robots [64]. Blossom addresses Sandoval et al.'s recommendations for fiction-inspired robots by being expressive, low resource through DIY, and by matching a user's investment through the platform's low cost [18]. Making Blossom accessible by involving users in its construction makes the robot more familiar and human while empowering users to become involved with robotics.

Generalizable outcomes and recommendations

- Making robots accessible through DIY and repairability familiarizes and humanizes their design.
- Involving users in the robot's construction empowers them to become involved in robotics.
- Enabling modularity and customization frames the robot as a canvas for users to critically reconsider what a robot *could* – rather than *should* – look like.
- Open-source and extensible designs remove the onus for others to redesign and reinvent existing platforms.

0.4.2 Design as Communication

Accessible robot design extends our capacity to communicate through the medium of robots (Figure 7). The ideated design is encoded through creation

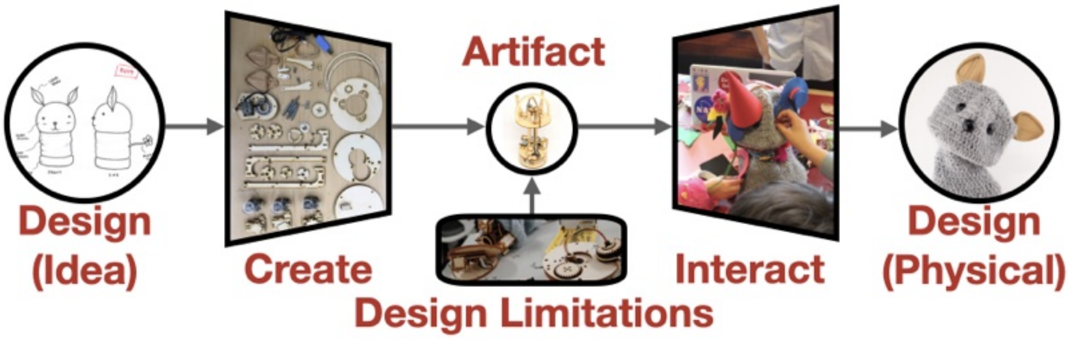


Figure 7: The communication model applied to design. The ideated design is encoded through creation into an artifact. The artifact is subject to noise in the form of design limitations. The artifact is decoded through interaction into a physical design.

into an artifact. The artifact is subject to noise in the form of design limitations, imperfections, and its spatiotemporal uniqueness; the noise is the design's "aura." The artifact is decoded through interaction into a physical design.

0.5 Movement

"Animation can explain whatever

the mind of man can conceive."

Walt Disney [65]

In this section, I provide an overview of Blossom's movement capabilities[6]. I describe the movement authoring system that comprises of a smartphone-based motion interface, which enabled crowdsourcing of emotive movement samples from lay users. I then describe the approach for expanding Blossom's behavior library using the crowdsourced dataset and generative neural networks. I

[6] My contributions include: designing the movement authoring system (assisted by Michael Hu); crowdsourcing the movements (assisted by Preston Rozwood); designing and implementing the neural networks (advised by Mason Bretan); performing user evaluations.

then detail two applications of this system – movement affective modification and face→movement translation – and the user evaluations performed. I then frame Blossom's movement as a form of robot-mediated communication: the human-crafted movement is transmitted through the medium of the movement generation model into a model-generated movement.

0.5.1 Related

Minimally expressive robots with limited audiovisual affordances express themselves through movement [66]. The existence of robots in physical reality sets them apart from voice- and screen-based agents. Even simple embodiments such as the Roomba [67] and an actuated stick [68] can convey emotions. Roboticists take cues from Disney's animation principles to create robots with lifelike movement [65], such as the Tofu robot which features a foam core to achieve squash-and-stretch [69]. These concepts are implemented in the passively smooth design of Blossom's tensile mechanisms. The movement authoring system is also accessible to crowdsource movement samples from a diverse range of users. These movement samples act as inputs to generative behavior neural network models for expanding the robot's behavior library.

0.5.2 Implementation

Movement authoring system

Robot movement authoring often requires specialized motion planning software and knowledge of robotics. Even accessible methods (e.g. learning

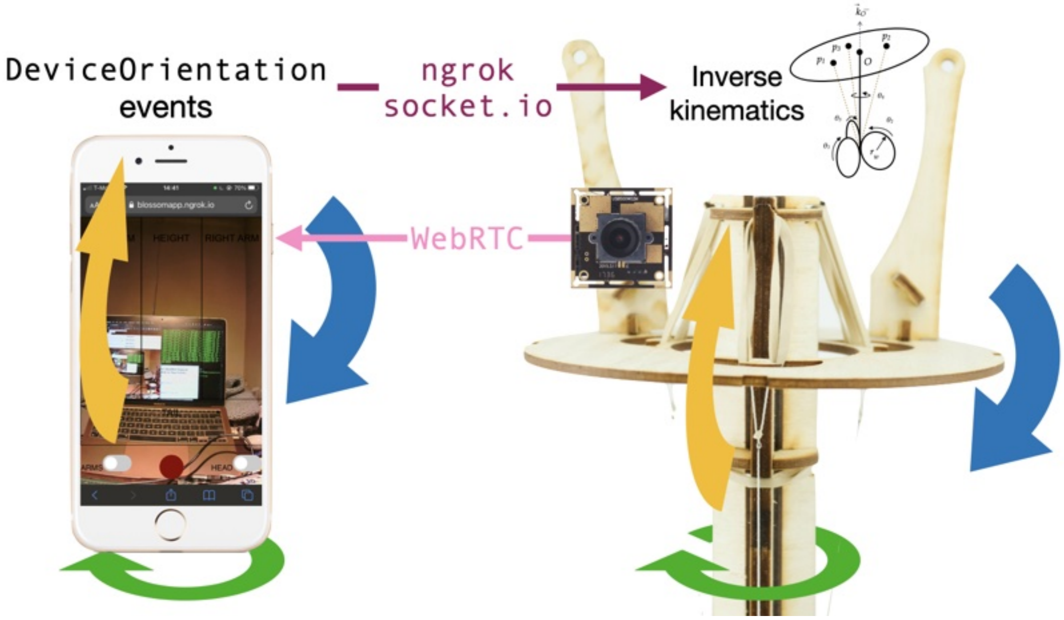


Figure 8: The movement authoring system. Users move the phone (left), and `DeviceOrientation` transmits the motion of the phone through `ngrok` and `socket.io` to the robot. The robot's back end inverse kinematics model calculates the motor positions required to match the phone's pose. For the telepresence application, `WebRTC` transmits a first-person video feed from a wide-angle camera embedded inside the robot's head to the phone interface.

from demonstration (LfD) [70]) require familiarity with a specific robot platform. In the interest of accessibility, we built the movement authoring system as a simple phone-based motion control interface (Figure 8, left). The `DeviceOrientation` API records the phone's motion, which passes through an inverse kinematics model to calculate the motor positions required for the robot's head to match the phone's pose. `ngrok` and `socket.io` handle communications between the phone and robot's control computer.

Data crowdsourcing and processing

We gathered movement samples from non-expert users. We first prompted users with videos of cartoon characters (e.g. Homer Simpson, SpongeBob

SquarePants, Pikachu) emoting either happiness, sadness, or anger, then instructed them to control Blossom as if it were conveying the same emotion. We recorded the emotive movements and yielded a dataset of approximately 150 movements (50 per each of happiness, sadness, and anger). The movement data stream is limited to 10 Hz and we chunked the samples into equal segments, between three to five seconds depending on the application.

Networks

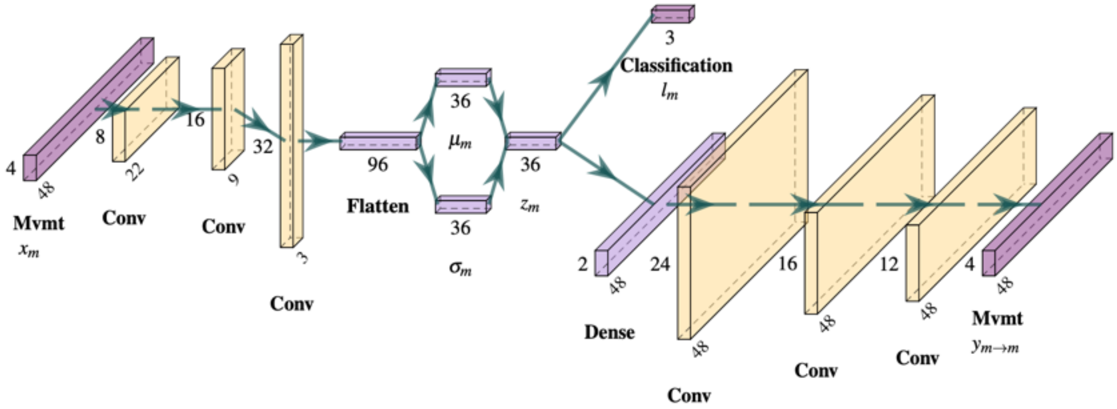


Figure 9: The movement VAE. The encoder compresses the input movement x_m into the latent embedding z_m (left). The decoder uncompresses the embedding into a reconstruction of the input $y_{m \to m}$ (right). The classifier separates the latent space by emotion label l_m (happy, sad, or angry) (top).

For our applications, we used a variational autoencoder (VAE) as the base neural network model [71]. A normal autoencoder (AE) passes input data (e.g. images, text) through an encoder into a compressed latent embedding space. The latent embeddings then pass through a decoder to reconstruct the original input data. The loss function to minimize (L_{AE}) is the difference between the input and its reconstruction, and passes through a differentiable scaling function f (e.g. absolute error, square error, etc).

$$L_{AE} = f(reconstruction - input) \tag{1}$$

A VAE extends the normal AE by adding the Kullback-Leibler (KL) divergence as a loss function L_{KL}. The KL divergence structures the latent space such that the embeddings z are sampled from a normal distribution $z \sim N(0,1)$. Our movement VAE takes as input movement samples x_m and outputs reconstructions $y_{m \to m}$ (Figure 9). We added an additional emotion classifier that operates on the movement emotion labels l_m and separates the latent embedding space by emotion classes (happy, sad, or angry). We can use the same dataset and underlying network structure for different applications, such as emotion modification and intermodal face→movement translation.

Emotion modification

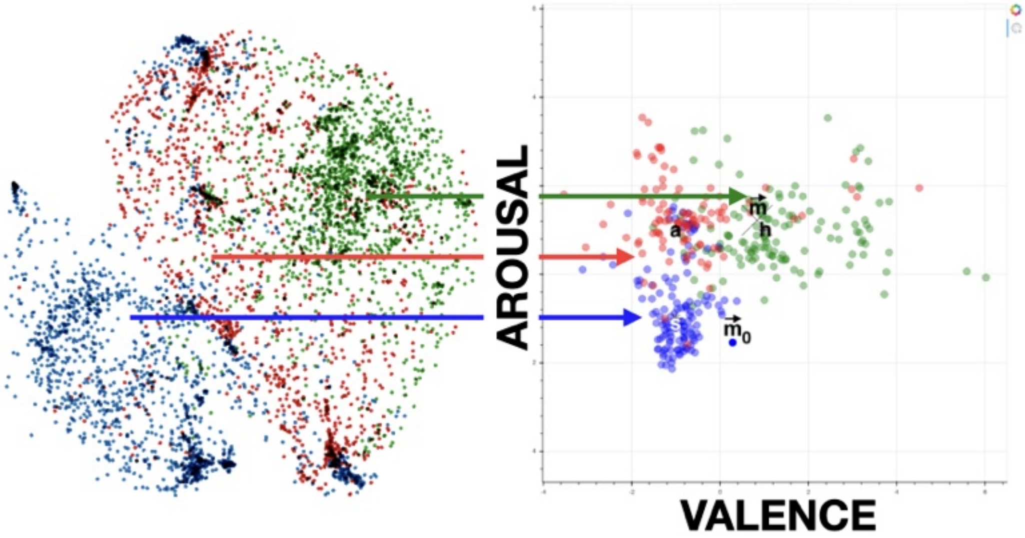


Figure 10: The movement modification latent embedding space and interface. After training the network and learning the latent space (left), we used linear regression to map the embeddings to the circumplex model of emotions (right) [4]. We can modify movements in the valence-arousal space by adding latent vectors to the embeddings and decoding through the network.

Drawing inspiration from neural modification for faces [72] and music [73], we sought to modify high-level movement features by modulating low-level parameters in the embedding space. After training the network and learning the latent representations, we used linear regression to map the latent space to the valence-arousal axes on the circumplex model of emotion (Figure 10) [4]. We can adjust the affective features by adding vectors in the latent space and decoding through the network, e.g. modifying happy movements into sad by decreasing both valence and arousal. We deployed a subjective user evaluation and found that the network could generate convincing movements, though only happy→sad and sad→angry sufficiently conveyed the correct target emotion.

Face→movement translation

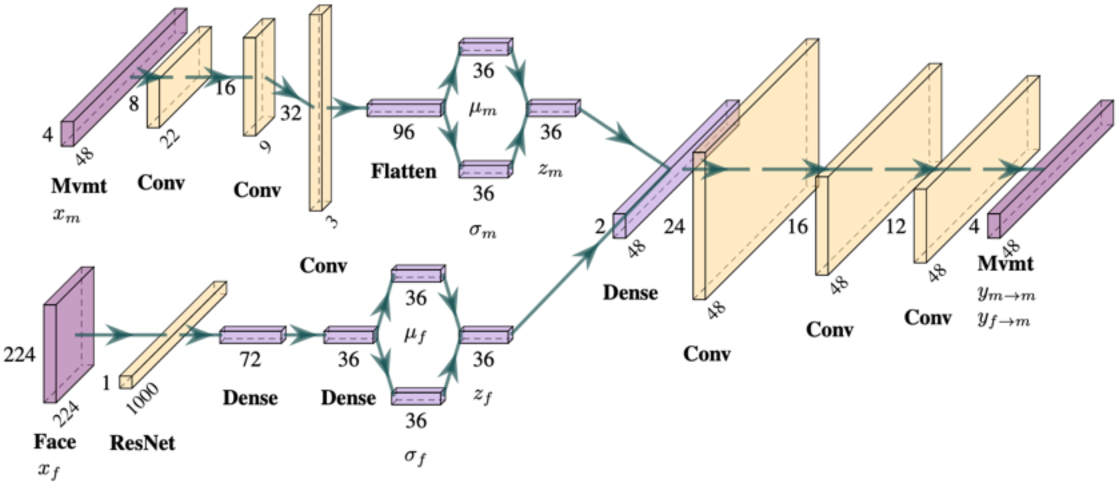


Figure 11: The face→movement translation network. The movement VAE remains intact (top left to right). An additional ResNet-based image encoder (bottom left) compresses images of facial expressions x_f into the shared latent space $\{z_m, z_f\}$. Once the end-to-end network is trained, we can translate faces into movements by passing images through the face encoder and movement decoder (bottom left to right).

Drawing inspiration from works in neural machine translation and image captioning [74, 75], we were interested in neural intermodal translation for movement. We chose facial expression images x_f from the Extended Cohn-

Kanade dataset due to its accessibility and use of the same labels as the movement data [76]. Because we lacked paired datasets, we used the supervised emotion labels as the semantic link between the disparate data modalities. We adjusted the base VAE network by appending an image encoder built on a pre-trained ResNet-50 model (Figure 11, bottom left) [77]. We adopted a technique from cross-modal manifold alignment [78] to align the movements embeddings z_m and face embeddings z_f in the latent space. We used a triplet loss function that attracts embeddings of the same emotion (e.g. happy movements attract happy faces and other happy movements) while repelling other emotions (e.g. happy movements repel sad and angry movements and faces). We deployed a user survey and found that the network could convincingly translate happy and sad movements (Figure 1, top right), but angry movements were not recognized. We attribute the difficulty of creating angry movements as an inherent limitation of the expressiveness of the limbless robot. We are experimenting with additional appendages (e.g. arms, tails) to enhance Blossom's expressiveness.

Generalizable outcomes and recommendations

- Making robot behavior authoring accessible (e.g. through familiar interfaces such as phones) enables crowdsourcing samples from non-expert populations, but requires secondary quality assurance.
- Using automated generation methods expands a robot's behavior library by complementing, but not supplanting, the human-crafted behaviors.
- Head-only robots may need more appendages, degrees of freedom, or expertly crafted movements to broaden their expressive range.

0.5.3 Movement as Communication

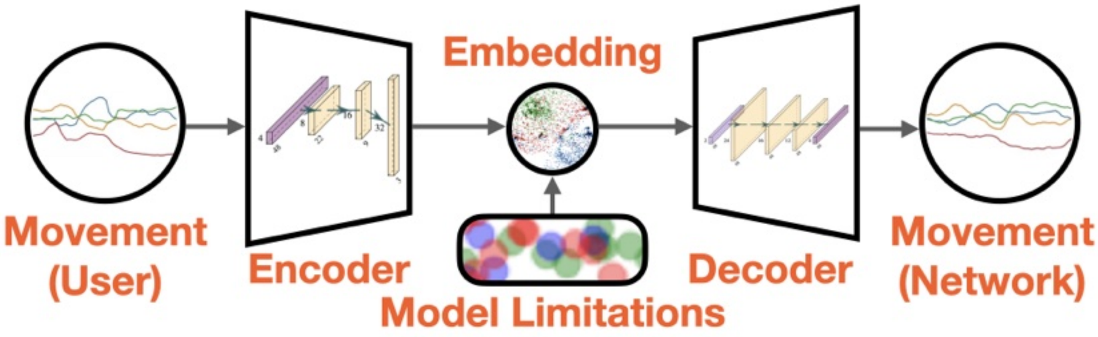


Figure 12: The communication model applied to movement. The movement is encoded through the network into a latent embedding space. The embeddings are subject to noise in the form of model limitations. The embeddings are decoded through the network into reconstructed or new movements.

Accessible robot movement authoring extends our capacity to communicate through the medium of robots (Figure 12). The human-crafted movement trajectory is encoded through the network into a latent embedding space. The latent embeddings are subject to noise in the form of model limitations and subjectivity of the data labels; the noise is the network's "aura." The latent embeddings are decoded through the network into reconstructed or newly generated movements.

0.6 Telepresence

"Virtual reality is simulacral, telerobotics is distal."

Ken Goldberg [79]

In this section, I provide an overview of Blossom's telepresence capabilities[7]. I describe the new functionality, including remote teleoperation and video monitoring for first- (looking *through* the robot) and third-person (looking *at* the robot) perspectives. I then detail the remote user evaluation of this system. I then frame Blossom's telepresence capabilities as a form of robot-mediated communication: the remote user's physical movement is transmitted through the medium of telecommunication into movement on the physical robot.

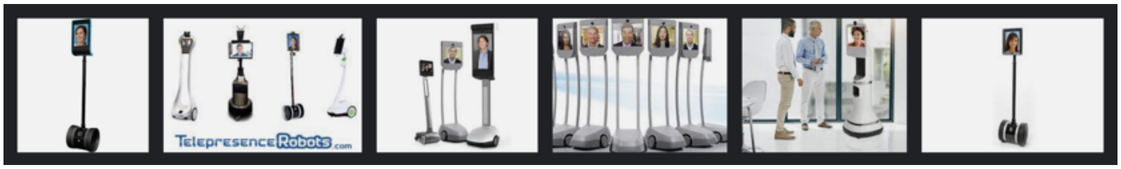


Figure 13: The first Google image results for "telepresence robot," revealing another stark aesthetic uniformity: tall pole-mounted screens on roving bases.

0.6.1 Related

Though telepresence robots are available as research platforms and commercial technologies (Figure 13), many are non-accessible due to cost or technological limitations, and often shunt the user's physical proprioception and embodiment by relying on button-and-joystick interfaces. Whether in a virtually stitched

[7] My contributions include: designing and implementing the technical capabilities to enable remote teleoperation; performing user evaluations.

space [80] or through a physical screen-based telepresence robot [81, 82], granting users motion-based agency of their remote embodiment improves communication and the experiential senses of co-location and agency [83]. Sakashita et al. used virtual reality systems to enable puppeteers to remotely control a robot [84]. Mandlekar et al. used smartphones to remotely crowdsource LfD trajectories for robot arm grasping tasks [85]. We drew inspiration from these works to create an accessible embodied robot motion control system.

0.6.2 Implementation

We extended the existing phone-based control system to enable teleoperation from a remote access point (Figure 1, bottom left). We embedded a wide-angle camera inside Blossom's head, which transmits a first-person video feed through `WebRTC` to the phone interface (Figure 8, left). An external webcam directed at the robot transmits an external third-person view of the whole robot to an optional desktop interface.

We have performed initial research evaluations of the telepresence system. We continued remotely crowdsourcing movements for the generative models, and also compared user preferences between the first- and third-person views. We found strong overall preferences for the third-person view, which may be attributed to Blossom's aesthetic appeal and the external perspective being more appropriate for the movement authoring task. Though some users noted difficulty in conveying disparate emotions, most users quickly adapted to the phone interface and expressed that the experience was "cool" and "(really) fun."

Generalizable outcomes and recommendations

- Motion-based telepresence robot control employs the user's own kinesthetic senses to heighten immersion in the remote environment.
- Screens as the principle design elements of telepresence robots should be reconsidered, especially in the context of mirror anxiety [86] and screen fatigue [87].
- Certain telepresence applications may benefit from a mix of first- and third-person perspectives.

0.6.3 Telepresence as Communication

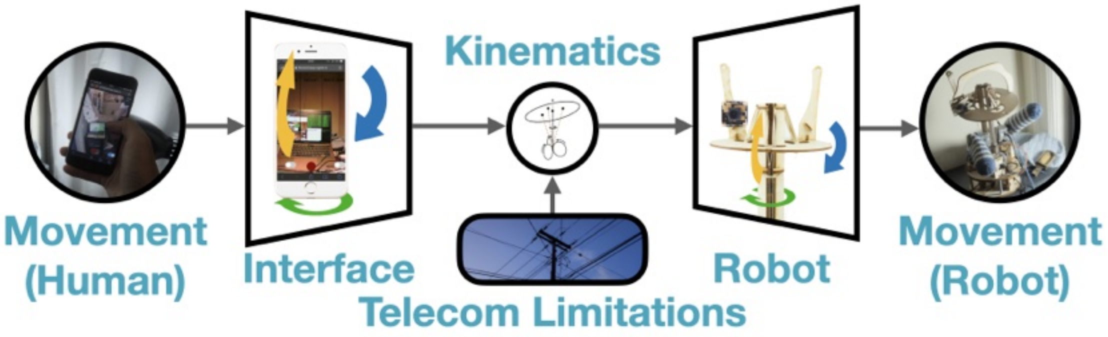


Figure 14: The communication model applied to telepresence. The remote user's physical movement is encoded through the phone interface into the kinematic definition of the motion. The kinematic definition is subject to noise in the form of telecommunication limitations. The kinematics are decoded through the mechanism into the robot's movement.

Accessible robot telepresence extends our capacity to communicate through the medium of robots (Figure 14). The remote user's physical movement is encoded through the phone interface into the kinematic definition of the motion. The kinematic definition is subject to noise in the form of telecommunication limitations such as bandwidth, lag, and mismatch between the theoretical and actual kinematics of the phone and robot; the noise is the telerobot's "aura" [88]. The kinematics are decoded through the mechanism into the robot's movement.

0.7 Discussion

The prior sections provided overviews of three phases of Blossom's development, and an interpretation of each as a form of robot-mediated communication. The following sections expand the phases and detail their constituent projects. These sections are comprised of the representative journal and conference publications, modified for continuity.

Part II

Design

CHAPTER 1

BLOSSOM: A HANDCRAFTED OPEN-SOURCE ROBOT

1.1 Introduction

The design of social robots with expressive capabilities is an active area of research in human-robot interaction (HRI). Dating back to MIT's Kismet—a robot specifically built to express internal states through facial movement and vocalizations [89, 7]—researchers in HRI have been developing methods for expressive capabilities of robots and collecting empirical evidence for the effects of these behaviors. Some of these robots use facial expressions [90, 91, 92, 93] while others express their internal states through bodily gestures [66, 94, 95, 96] or other modalities [97, 98]. More recently, consumer electronics companies have also started to explore expressive social robots as commercial products [99, 100, 101].

Designing and building such a robot, however, requires extensive knowledge and resources in mechanical and electrical engineering. Similarly, designing and implementing the robot's expressive gestures and behaviors requires professional skills in computer science and 3D character animation. This makes robot building and programming inaccessible to a large swath of users.

This lack of accessibility limits the use of social robots for both researchers and end-users. For example, most researchers in HRI have a choice of one of a handful of programmable social robots, such as the Softbank Robotics's NAO or Pepper robots, Philips's iCat, Rethink Robotics's Baxter, or the MyKeepon platform. These robots are subsequently exceedingly prevalent in the HRI liter-

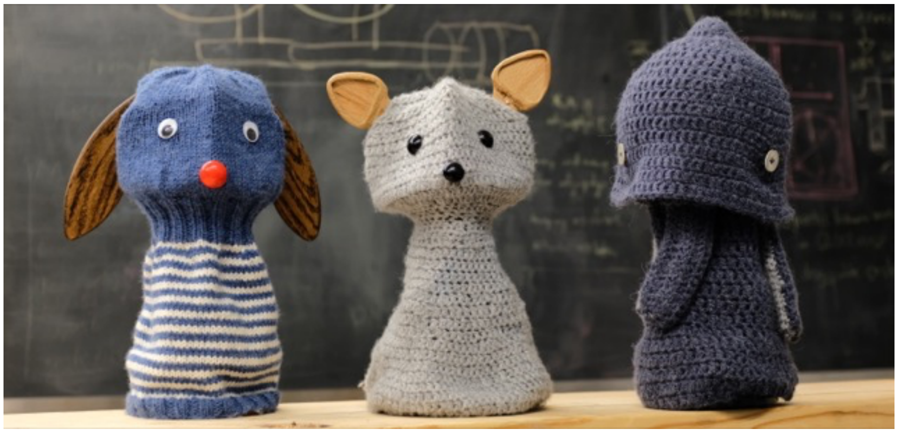


Figure 1.1: Three variations of Blossom with different embodiments and accessories. The robot on the left is knitted, and the two robots on the right are crocheted. The two robots on the left display swappable wooden ears and a number of attachable facial features, while the robot on the right features soft silicon arms as appendages.

ature, e.g. [102, 103, 104, 105, 106, 107, 108, 109]. Each of these robot has a single outward appearance, which is overcome at times with adornments such as hats or other accessories [110]. Still, it is difficult to adapt the robot's appearance to the task at hand, rendering them inflexible with respect to specific applications and personalization.

The majority of social robots are also rigid in a more literal physical sense: Their exteriors are made of hard plastic or metal shells manufactured using additive and subtractive methods such as 3D printing, molding, and milling. These exteriors are fixed to direct or geared drive mechanisms and rigid linkages with fasteners such as bolts and adhesives to form solid connections. This mechanical rigidity restricts the robot's expressiveness and interactive capabilities. Rigid actuation mechanisms make it difficult to achieve smooth, organic movement without complicated software control or trajectory generation. Stiff direct linkage mechanisms also discourage physical interaction due to their hard tactile affordance and fear of damaging internal components.

In this work we present Blossom, an open-source robotics platform for researchers and hobbyists, with the goal of addressing the issues identified above. Blossom is designed to allow researchers and end-users to imagine and build their own robot, enabling more flexible design possibilities in the robot's appearance, structure, and behaviors. This could increase adoption and help diversify HRI research. In addition, Blossom offers a novel mechanical design with compliant, organic movement in mind, to support expressiveness and interactivity.

Blossom thus attempts to achieve three design objectives: *accessibility*, *flexibility*, and *expressiveness*, implemented through the following design choices:

- The robot can be easily put together by lay-users.
- It has modifiable degrees-of-freedom (DoFs), but is still predictably expressive.
- It uses a tensile mechanical structure that affords smooth movements and safe interaction.
- Its appearance can be handcrafted with traditional crafts.
- Both its mechanism and exterior can be made from readily available low-cost materials.
- New behaviors can be defined without requiring programming or computer animation skills.
- Behaviors are accessible through an open interface suitable for a broad range of applications.

Notably, Blossom is not a robotics kit in the same vein as LEGO Mindstorms™ or Meccano™, which differ in two important ways. First, these kits provide a widely open-ended design space which is not tailored to any specific application. In particular, they are not designed with expressive behavior or social interaction in mind. Second, these kits cater mostly to technically-oriented users and focus on the robot's construction rather than its use. In contrast, Blossom is socially-oriented, while still being easily customized by non-technical users, and focused on the end-user of the robot.

As a use case, we imagine a social science research group with limited technical expertise but interest in a research question related to HRI. Researchers in this group should be able to quickly build, fashion, and use a Blossom robot and define behaviors specific to their application. Another scenario could be a lay-user who is uninterested in engineering and programming but wants to build a social robot for their personal use with a particular appearance and set of behaviors.

In this paper we present Blossom's mechanical, electronics, and software implementation and detail the customizable exterior and behavior of the robot. To evaluate the design, we provide four case studies of field deployments where users implemented or interacted with Blossom robots.

1.2 Related Work

Blossom relates to the existing literature in social robot design, gesture generation, and open-source robotics construction kits.

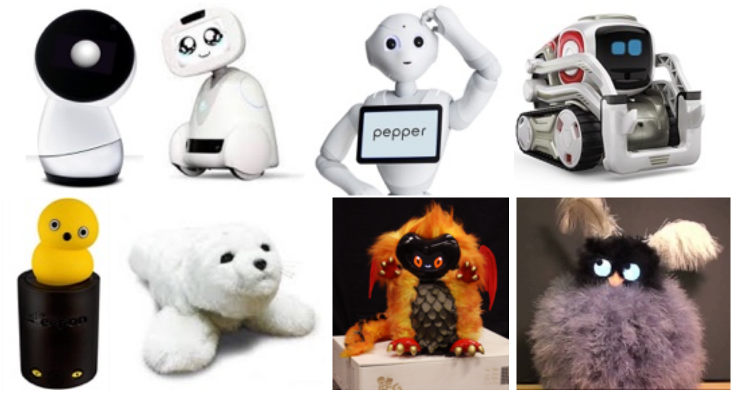


Figure 1.2: Jibo, Buddy, Pepper, and Cozmo (top) are examples of social robots with similar design features related to consumer electronics devices. Keepon, Paro, DragonBot, and Tofu (bottom) exhibit softer and more zoomorphic embodiments.

1.2.1 Social Robot Design

Aesthetic designs for social robots range from product-like to organic. Jibo, Buddy, Pepper, and Cozmo (Figure 1.2 top) are examples of robots with features akin to those of consumer electronics devices, such as straight lines, rounded edges, touch screens, and illuminated accents [22, 111, 112, 113]. On the other side of the spectrum are creature-like robots such as Keepon, Paro, DragonBot, and Tofu (Figure 1.2 bottom), evoking a more zoomorphic aesthetic [114, 115, 116, 69]. All these robots' appearance and DoFs are fixed and not customizable by their users.

The choice of materials also plays an important role in robot design. Appliance-like robots are generally made from rigid materials such as plastics or metals with smooth finishes. While the use of alternative and handcrafted materials has been emergent in other interactive technologies [117, 118], it has been less explored in social robotics. OPSORO (Open Platform for SOcial RObots) is an exception in that it uses fabrics in the design of its soft covers [119]. Additional examples exist in hobbyist circles, such as TJBot, a single-DoF desktop robot, and Smartibot, a phone-controlled mobile robot, both constructed from cardboard [120, 121].

For actuation, most robots use rigid mechanisms and direct-drive motors to achieve movement. Smooth motions must thus be achieved through intricate control software and trajectory generation tailored to the robot's kinematics. Some have explored pneumatic actuators that can achieve smooth motion through mechanical design [122], but the pumps and compressors required to drive these systems are often noisy and cumbersome. Another approach is to use tensile mechanisms to trade precise control for range and smoothness of motion. One example is the prototype robot Tofu, which has a head attached to a foam column with cables pulling on the head for actuation [69]. Another example is Probo, a robot with a tensile trunk [123].

Compared to these social robots, Blossom's design differs in that it is flexible, inside and out. Blossom features an open-ended exterior meant to be customized by end-users through handcrafted materials, and its interior actuation mechanism uses compliant tensile components. This actuation mechanism is kinematically similar to Stewart platform mechanisms, which were used in the DragonBot [116] and Peeqo [124] robots. However, in contrast to those mechanisms, Blossom uses compliant components to achieve smooth and lifelike movement without requiring complicated software control. It also achieves a larger range of motion than Stewart platforms with only half the number of motors. Blossom's mechanism bears similarity to that of the Tofu robot but has a larger range of motion due to its free-floating platform; it is also simpler to manufacture. In addition, Blossom's exterior cover and internal mechanisms are not affixed to each other, allowing for more expressive movement through slip and secondary action.

1.2.2 Robot Gesture Generation

Generating smooth and natural movements and gestures for social robots can be a lengthy and complicated process. Traditional methods for gesture generation are generally programmatic, require knowledge of the robot's kinematics, and are not accessible to novice users. Allowing users to create their own gestures affords a novel method of personalizing the robot and could help mitigate the novelty effect stemming from robot movements being repetitive and predictable.

In efforts to make robot gesture generation more accessible and intuitive, researchers have developed methods involving physical manipulation of the robot. Mirror puppeteering involves placing markers on parts of the robot and manipulating it in front of a camera to record movements [125]. Robots like Topobo and ChainFORM implement "kinetic memory" which allows gestures to be recorded by physically moving the robot's appendages by using back-drivable motors with position encoders [126, 127]. Learning from Demonstration supplements either approach by having the user provide corrective demonstrations to iteratively teach the proper movements [70]. These approaches are more intuitive than programmatic methods, but make it difficult to perform full gesture generation in real-time, often requiring a layered approach in which each DoF is actuated one at a time. In some cases, keyframes and interpolation are used to complement the puppeteering activity. This approach makes it hard to achieve high-quality expressive movements.

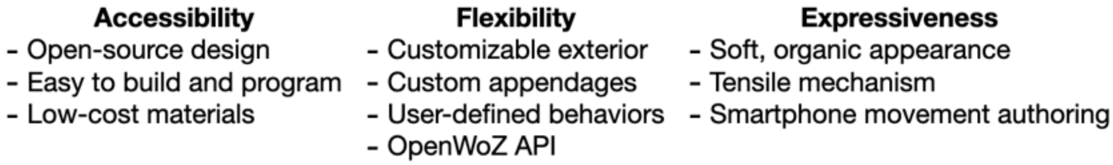
In contrast, Blossom allows lay-users to create gestures using a smartphone as a puppeteering interface. The robot's actuation mechanism kinematically resembles a free-floating platform and is controlled by mapping the orientation of the phone to that of the robot's head platform directly, enabling real-time exploration and recording of gestures.

1.2.3 Open-Source Robots

There are a few existing open-source robotics projects that allow users to build their own robot from openly accessible online data files. Robots like iCub, Poppy, and InMoov are examples of open-source platforms that have humanoid bodies and intricate mechanical and software designs [128, 129, 130]. Non-anthropomorphic open-source robots such as Hexy and TurtleBot are comparatively simpler [131, 132], owing to their more abstract embodiments. While the design of these robots are openly accessible, their appearances are largely fixed, and the systems require a high degree of technical knowledge to build, program, and use. Some of these robots can be programmed through visual block-based languages such as Scratch or Blockly [133, 134], but this programmatic approach does not support the authoring of new expressive gestures, making them ill-adapted for social robotics applications.

Among open-source robot platforms, OPSORO is specifically socially-oriented. It is comprised of modular components representing different facial features and a customizable exterior cover that is made from soft materials [119]. This makes it more accessible and expressive than most other open-source social robots. A semester-long deployment of the robot in a student design course produced several unique embodiments ranging from animals to the likeness of Albert Einstein. That said, OPSORO was largely designed for facial expressions, and its behaviors must be defined programmatically.

1.3 Design Objectives



Accessibility	Flexibility	Expressiveness
- Open-source design	- Customizable exterior	- Soft, organic appearance
- Easy to build and program	- Custom appendages	- Tensile mechanism
- Low-cost materials	- User-defined behaviors	- Smartphone movement authoring
	- OpenWoZ API	

Figure 1.3: Design objectives of the Blossom platform and features that address these objectives.

Blossom, in contrast, is designed to allow lay-users to create their own robot end-to-end, from building its structure, through the design of its appearance, to the authoring of new gestures and the combination of these gestures into behaviors. It address the gaps identified in existing social robot design by addressing three design objectives (Figure 1.3):

Accessibility Lay-users without technical knowledge should be able to contribute to all aspects of building and programming the robot.

Flexibility The robot's design should allow end-users to alter aspects of its appearance, mechanical structure, and interactive capabilities.

Expressiveness Despite the accessibility and flexibility of the robot's design, it should maintain a high degree of expressiveness in its appearance and movement. The movement should be smooth without relying on complicated control software.

1.4 Implementation

This section describes the technical implementation of Blossom in pursuit of the above objectives. It serves to enable the replication and extension of the technical aspects of the robot design. In overview, the robot's mechanical structure is made up of flat components which can be cut from sheets of wood or acrylic and uses snap and press fits to reduce the need for fasteners. It is actuated by a non-rigid tensile mechanism constructed from elastic components to achieve compliant, organic movement. One of the DoFs is open-ended and can be used to actuate custom appendages. The electrical design uses mostly snap connectors that do not require soldering, and allows the robot to be either controlled by an external computer via USB or run untethered using an on-board battery-powered microcomputer. In both cases, an open Hypertext Transport Protocol (HTTP) application programming interface (API) allows remote control and programming of the robot's behaviors. The robot's gestures are authored using a smartphone-based puppeteering application which can be recorded and played back in real-time during operation, or saved on the robot to be triggered by the remote HTTP API.

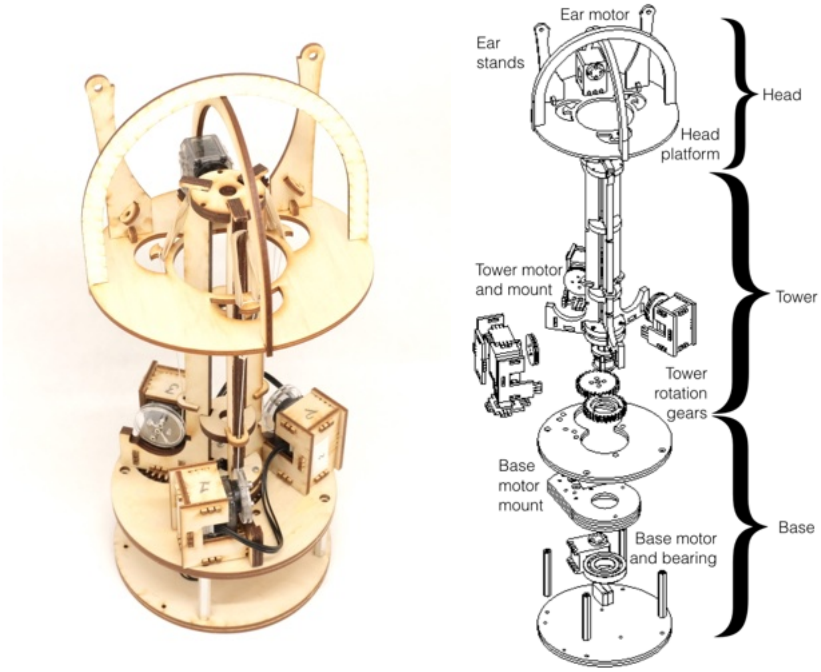


Figure 1.4: The inner tensile actuation mechanism and exploded view. The main expressive element is the head platform which is suspended from a tower by rubber bands and actuated by cables driven by motors at the bottom of the tower. The tower itself is rotated by the base motor. As an example of an appendage, the head platform features ear stands and a motor for actuating the ears.

1.4.1 Mechanics

Blossom's mechanical design is centered around a free-floating "head" platform, which is actuated using a tensile mechanism for power transmission (Figure 1.4). The head is suspended from the top of the central tower structure with rubber bands and is actuated by reeling in cables from the bottom of the tower. The design is related to the Stewart platform mechanism which has been used in other social robots [135, 124], but Blossom's design is non-rigid and allows for a larger range of motion than a Stewart platform, all while reducing the number of motors from six to three. This is made possible through the variable lengths of the tensile components, whereas a Stewart platform is limited by the fixed lengths of its rigid linkages.

This actuation mechanism also bears similarity to the one used in the prototype Tofu robot [69]. While there is not much published information about the robot, it is described as also using an elastic element (a cylindrical foam core) to hold a head which is actuated by cables. However, unlike the foam core used in Tofu to which the head and skin are rigidly attached, Blossom uses a free-floating head with elastic bands (Figure 1.8 left), as well as a freely moving exterior cover. This not only lowers the cost and difficulty of assembly, but also allows for larger range of motion that is accentuated by secondary motions. Additional movement is produced by a fourth motor in the base to rotate the tower assembly and a fifth motor on the head platform that actuates customizable appendages.

Range of Motion

Figure 1.5 shows examples of the head platform's range of motion. The gestures of the inner mechanism can be classified as superpositions of several basic motion primitives: moving all the tower motors synchronously causes vertical translation, asynchronous motion results in pitching or rolling, and moving the base motor produces yawing. These fundamental motions are combined in timed sequences to create expressive gestures.

In addition to the increased range of motion, the tensile mechanism affords gestures that are smooth and organic-looking to an extent that would be challenging to replicate through software alone. The physical elasticity specifically supports several principles of animation [136, 137]. The cables and elastic bands

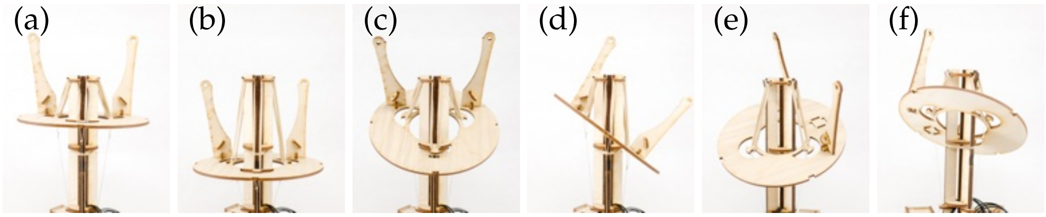


Figure 1.5: Examples of the mechanism's range of motion. Vertical translation (a→b) and rotations (c, d) are combined to create more complex gestures (e, f).

provide a springiness that enables ease-in and -out in smooth arcs. The variable lengths of these components allow for greater exaggeration in motion. The momentum of the platform during quick movements elicits natural secondary motions such as overshoot and oscillation that would otherwise necessitate complex trajectory generation in motion planning software.

1.5 Kinematics

The novel design of Blossom's internal mechanism requires custom kinematics for gesture generation and simulation.

1.5.1 Forward Kinematics

Figure 1.6 shows an approximation of the inner mechanism. For simplification, the elastic bands are neglected and cables are assumed to be rigid links of variable length capable of both pushing and pulling the platform. The attachment points of the cables are denoted p_{1-3}. As shown in Figure 1.6(a), the tower motor wheels of radii r_w rotate by θ_{1-3} and the base motor rotation about the vertical axis is denoted by θ_4.

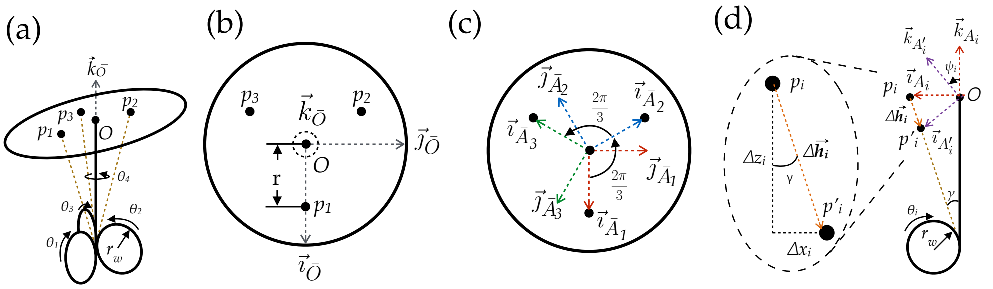


Figure 1.6: Kinematic diagrams of the robot's inner mechanism. As shown in (a), the inertial reference frame $\bar{O} =< \vec{i}_{\bar{O}}, \vec{j}_{\bar{O}}, \vec{k}_{\bar{O}} >$ is defined with the origin O at the center of the platform when at rest. The lines from the base of the tower to the attachment points p_{1-3} represent the cable. The tower motors that actuate the platform are at the base of the tower and rotate the motor wheels of radius r_w by angle θ_{1-3}. The base motor located below the tower motors (not depicted) rotates the tower about the vertical axis by θ_4. Top view diagrams in (b) show the locations of the attachment points. The frames $\bar{A}_1$, $\bar{A}_2$, and $\bar{A}_3$ depicted in (c) are aligned with the attachment points and rotate about the vertical $\vec{k}_{\bar{O}}$ axis shown in (b). The side view (d) shows the actuation mechanics of a single attachment point. The frame $\bar{A}'$ is aligned with $\bar{A}$ as it rotates about the vertical $\vec{k}_{\bar{O}}$ axis, but additionally rotates about the shared $\vec{j}_{\bar{A}} = \vec{j}_{\bar{A}'}$ axis out of the page. This results in the rotation from $\bar{A}$ to $\bar{A}'$ by the angle ψ_i. The displacement is approximated by $\Delta\vec{h}_i$ with components Δx_i and Δz_i in the $-\vec{i}_{\bar{A}}$ and $-\vec{k}_{\bar{A}}$ axes, respectively. The angle γ is the angle between the vertical axis and the line formed by the cable when the platform is at rest.

Top views in Figures 1.6(b) and (c) depict the locations of the attachment points and define the intermediate frames $(\bar{A}_1, \bar{A}_2, \bar{A}_3)$. These intermediate frames are aligned with the attachment points p_{1-3} respectively and rotate about the vertical inertial axis, with all $\vec{k}$ axes shared: $\vec{k}_{\bar{O}} = \vec{k}_{\bar{A}_1} = \vec{k}_{\bar{A}_2} = \vec{k}_{\bar{A}_3}$, and shown as $\vec{k}_{\bar{O}}$ in Figure 1.6(b).

We are interested in the pose of the head platform given a set of motor angles θ_{1-4}. Consider the movement of one of the attachment points, p_i as depicted in Figure 1.6(d). The rotation of the motor wheel of radius r_w by angle θ_i causes the cable to be pulled in by length $r_w\theta_i$. Denoting the angle between the vertical axis and the cable as γ, this shortening of the cable results in the displacement $\Delta\vec{h}_i$ of point p_i from its resting position to the actuated point p_i':

$$\Delta\vec{h}_i = -\Delta x_i \vec{i}_{\bar{J}} - \Delta z_i \vec{k}_{\bar{J}} = -r_w \theta_i \sin\gamma \vec{i}_{\bar{J}} - r_w \theta_i \cos\gamma \vec{k}_{\bar{J}} \tag{1.1}$$

A simplifying assumption is made that the attachment point moves along this line and that γ remains constant. The resulting actuated reference frame $\bar{A}_i{}'$ is a rotation of the original $\bar{A}_1$ about the shared $\vec{j}_{\bar{A}_1} = \vec{j}_{\bar{A}_i{}'}$ axis out of the page. If we denote the vectors from O to the resting position of the attachment point p_i as $\vec{r}_i$, we get $\vec{r}_i = r\vec{i}_{A_i}$. After actuating motor i, we get the new vector from O to p_i, $\vec{r}_i'$:

$$\vec{r}_i' = \vec{r}_i + \Delta\vec{h}_i = r\vec{i}_{A_i} + \Delta\vec{h}_i \tag{1.2}$$

These vectors need to further be transformed to the inertial frame by the planar rotation matrices of θ_4 for $\bar{A}_1$ and of $\theta_4 + \frac{2\pi}{3}$ and $\theta_4 + \frac{4\pi}{3}$ for $\bar{A}_2$ and $\bar{A}_3$, respectively. The calculated positions of the attachment points can then be used to determine the resulting orientation of the platform.

To do so, we define unit normal vectors for the idle and transformed orientations as $\vec{N}$ and $\vec{N}'$ respectively. We take $\vec{N} = \vec{k}_{\bar{O}}$ to be simply pointing upwards from O. The transformed vector $\vec{k}_{\bar{O}}$ can be calculated from a normalized cross product of the transformed attachment point vectors in the plane of the actuated platform.

$$\vec{N}' = \frac{(\vec{r}_1' - \vec{r}_2') \times (\vec{r}_1' - \vec{r}_3')}{|(\vec{r}_1' - \vec{r}_2') \times (\vec{r}_1' - \vec{r}_3')|} \tag{1.3}$$

The normal vector to the rotation plane $\vec{M}$ can be calculated and used to determine the quaternion rotation angle α and frame defined in $\vec{v}$.

$$\vec{M} = \frac{\vec{N} + \vec{N}'}{|\vec{N} + \vec{N}'|} \tag{1.4}$$

$$\alpha = \vec{M} \cdot \vec{N} \qquad \vec{v} = \vec{M} \times \vec{N} \tag{1.5}$$

$$\vec{q} = \begin{bmatrix} \alpha \\ \\ \vec{v} \end{bmatrix} \tag{1.6}$$

This quaternion is then used to determine the change in orientation and the downward displacement is approximated using Horn's method. The resulting changes in position and orientation are superimposed to determine the final pose.

1.5.2 Inverse Kinematics

Given the above forward kinematics solution, we can compute the head platform orientation given known motor positions. The same model can also be used to derive the inverse kinematics to calculate the required motor positions to achieve a desired final orientation of the platform. First, the Euler angles (ψ,θ, and ϕ about the body $\vec{i}_{\bar{B}}-$, $\vec{j}_{\bar{B}}-$, and $\vec{k}_{\bar{B}}-$axes, respectively) of the desired orientation are used to derive the rotation matrix ${}^{\bar{O}}R^{\bar{B}}$ from the $\bar{B}$ frame in the final orientation to the inertial frame $\vec{O}$:

$$^{\bar{O}}R^{\bar{B}} = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - c\phi s\psi & s\psi s\phi + c\psi c\phi s\theta] \\ c\theta s\psi & c\psi c\phi + s\psi s\theta s\phi & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{1.7}$$

The rotation matrix is used to transform the representations of the positions of the attachment points $\vec{r}_{p'_i}$ from the body frame $\vec{B}$ to the inertial frame $\vec{O}$:

$$\{\vec{r}_{p'_i}\}_{\bar{O}} =^{\bar{O}} R^{\bar{B}}\{\vec{r}_{p'_i}\}_{\bar{B}} \tag{1.8}$$

From the initial ($\vec{r}_{p_i}$) and transformed ($\vec{r}_{p'_i}$) positions of the attachment points, the displacements $\Delta\vec{h}_i$ can be calculated by:

$$\Delta\vec{h}_i = \vec{r}_{p'_i} - \vec{r}_{p_i} \tag{1.9}$$

Given the known size of the motor wheel r_w we can then calculate the angular motor displacement θ_i:

$$|\Delta\vec{h}_i| = r_w\theta_i \rightarrow \theta_i = \frac{|\Delta\vec{h}_i|}{r_w} \tag{1.10}$$

Fabrication

Blossom's fabrication process relies almost exclusively on laser cutting, which has advantages over 3D printing for its reproducibility and speed, as well as for the affordance of low-cost, recyclable, and readily available materials such as wood and cardboard. The structure uses snap fits similar to OPSORO's design

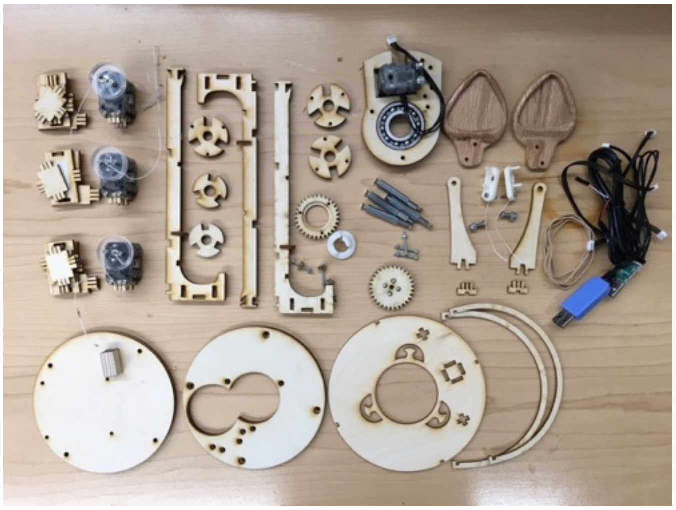


Figure 1.7: Layout of the components used to assemble the mechanism.

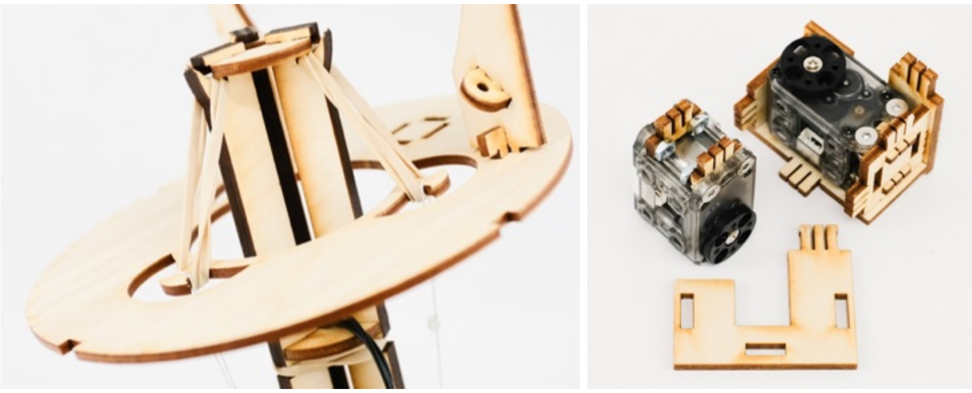


Figure 1.8: Detail of the compliant components (elastic bands and strings) used to suspend the head platform (left) and a snap-fit motor mount (right). Snap and press fits are used throughout the structure for ease of assembly and to reduce the amount of required hardware.

to reduce the amount of required hardware fasteners while being expandable with different appendages and motor configurations. Figure 1.7 shows all of components needed to build one Blossom robot with ears as appendages. Figure 1.8 (right) shows the motor mount as an example of a snap-fit component.

1.5.3 Electronics

The electronics system also supports the design principle of accessibility by consisting of commercially-available components that use simple mechanical connectors, reducing the need for soldering.

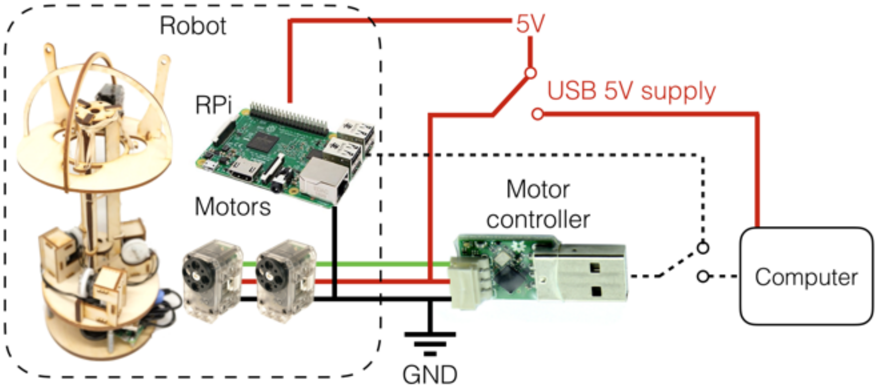


Figure 1.9: Electrical component diagram. The robot can be used both in self-contained mode through an internal system-on-board, or controlled by an external computer. The motors within the robot are daisy-chained and thus only require one connection to the computer via the USB motor controller.

Figure 1.9 shows the components of the robot's electronics system. The robot consists of five daisy-chained servo motors and a Raspberry Pi (RPi) microcomputer running the Linux operating system. The motors are controlled by the computer via a USB motor controller, which contains hardware to translate the USB protocol to Transistor-Transistor Logic (TTL) signals, and manages the half-duplex communication protocol of the servo motors.[1]

The robot can be used in one of two modes: self-contained or externally controlled. In the self-contained mode, the motors are connected to the RPi with the motor controller. Both the RPi and motors are powered by a 5-Volt (5V) power source such as a portable battery pack, but separate power connectors are

[1] The motors are Dynamixel™XL-320 and the USB motor controller is either a Xevelabs™USB2AX USB-to-TTL interface or a Dynamixel™U2D2.

used to prevent current overload on the logic components.

In the externally controlled mode, the RPi is unused and the motor controller is plugged into an external UNIX-based computer. Because power cannot be supplied through the motor controller and to prevent overcurrent on the computer's USB port, the motors must be powered from a separate 5V source such as an additional USB port or an external power supply.

1.5.4 Software

The software system of Blossom supports the objectives of flexibility and accessibility. The same software runs whether the robot is run in self-contained mode or externally controlled with a computer. The software uses the OpenWoZ framework [138], allowing for flexibility in application by exposing each of the robot's behaviors to an HTTP Universal Resource Identifier (URI)-based interface. This provides a flexible interface for creating behaviors, from high-level control of the robot to programming it in code.

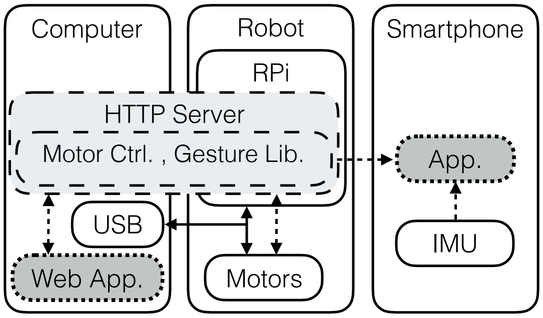


Figure 1.10: Combined hardware and software diagram. Solid lines denote hardware, dashed lines and light gray shading denotes back-end software, and dotted lines with dark gray shading denotes user interfaces. Physical connections are denoted by solid arrows and software communication is denoted by dashed arrows.

The robot's software is made up of three main components (Figure 1.10): a motor control module and gesture library to command the motors as well as to store and play back authored movements (shaded light gray); an HTTP server which listens to incoming requests and activates the appropriate gestures (shaded light gray); and the various user interfaces (UIs) for commanding the robot (shaded dark gray).

Motor Control Module and Gesture Library

The motor control is built on top of the PyPot motor control library [129], which abstracts the low-level serial communication for the servo motors to higher-level commands such as addressing motors and setting goal positions and speeds.

In the motor control module, robots are defined by the motors used and their respective ranges. The motors can be commanded directly, or controlled by executing gestures from a library. Gestures are stored as timed sequences of positions for each motor on the robot. The gestures can be played back with modulations to the speed, range, or posture.

HTTP Server

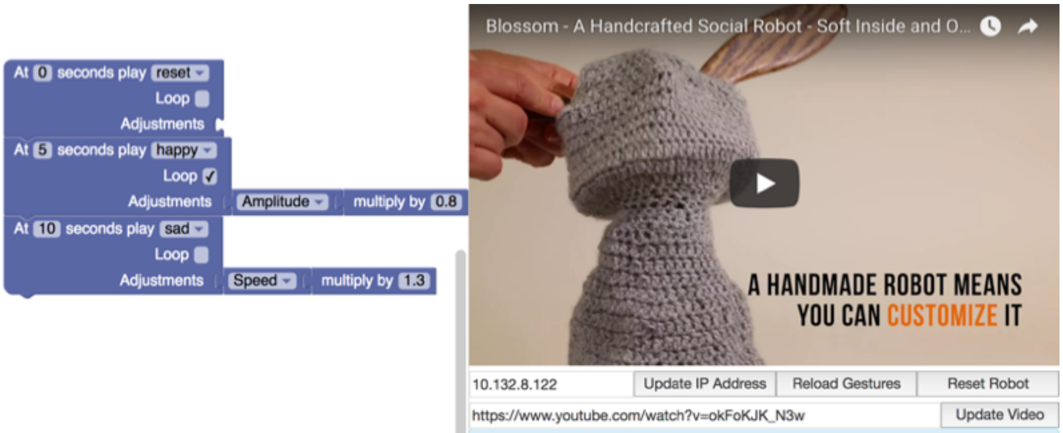
The control computer includes an HTTP server that enables Representational State Transfer ("RESTful") communication with the robot, allowing for the robot to be commanded from any device on the local network. This enables an open-ended method for interfacing with the robot and makes it easy to build Wizard-of-Oz (WoZ) interfaces, create custom applications that use sensor information, or communicate with existing web-based services or Internet-enabled devices.

The RESTful API receives the desired command or gesture and modulation parameters. For example, to play back a gesture titled "nodding" at 0.8 times the recorded speed and 1.4 times the amplitude of the original range of motion, the REST command would be `/s/nodding?speed=0.8&amp=1.4`. Examples of other functions include retrieving a list of available gestures and commanding the robot to a given position. This implementation follows the modular command structure of OpenWoZ [138] and affords flexible communication between the robot and clients built into user interfaces.

Additional behaviors can be added to the open-source HTTP server simply by defining a function and linking it to a RESTful command. Parameters are passed to the function as a URI string, and the custom behavior can parse the parameters. This is, for example, how different "breathing" and other programmatic idle behaviors are implemented.

User Interfaces

We demonstrate the flexibility afforded by the software architecture by presenting several methods we have developed to control the robot. In the simplest case, users can use the command line interface (CLI) on the terminal that started the robot HTTP server to trigger any command available to the RESTful API by simply typing in the REST URI. Beyond the CLI, we developed web and smartphone WoZ applications for high-level operation of the robot. A "soundboard" design enables the creation of buttons for triggering gestures, or for gesture/modulation combinations (Figure 1.12 (b)).


At 0 seconds play reset
Loop ☐
Adjustments
At 5 seconds play happy
Loop ☑
Adjustments Amplitude multiply by 0.8
At 10 seconds play sad
Loop ☐
Adjustments Speed multiply by 1.3

10.132.8.122
https://www.youtube.com/watch?v=okFoKJK_N3w

Figure 1.11: The web application used to trigger gestures timed to a video. The Blockly interface is used to denote when to trigger gestures and how to modify playback speed, amplitude, posture, or looping. In this example, the robot resets at the beginning of the video, plays the "happy" gesture at 5 seconds at 0.8 times the original amplitude (range of movement) and loops until 10 seconds, at which it then plays the "sad" gesture sped up by a factor of 1.3.

An additional web application is embedded in a web page (Figure 1.11). It allows Blossom to "react" to an online video as part of a research project in our laboratory, in which Blossom acts as a video-watching companion. The web page includes a video player and a Blockly interface for triggering gestures at specified timestamps and modulating them, allowing users to easily choreograph movement sequences to videos.

The mobile application (Figure 1.12) also supports triggering and modulating gestures but, more importantly, utilizes the phone as a puppeteering device to control the robot's expressive elements. Using smartphones as an input device supports the accessibility design objective by allowing lay-users to easily create behaviors for the robot without having to manually program its movements.

The puppeteering system leverages the smartphone's built-in inertial mea-

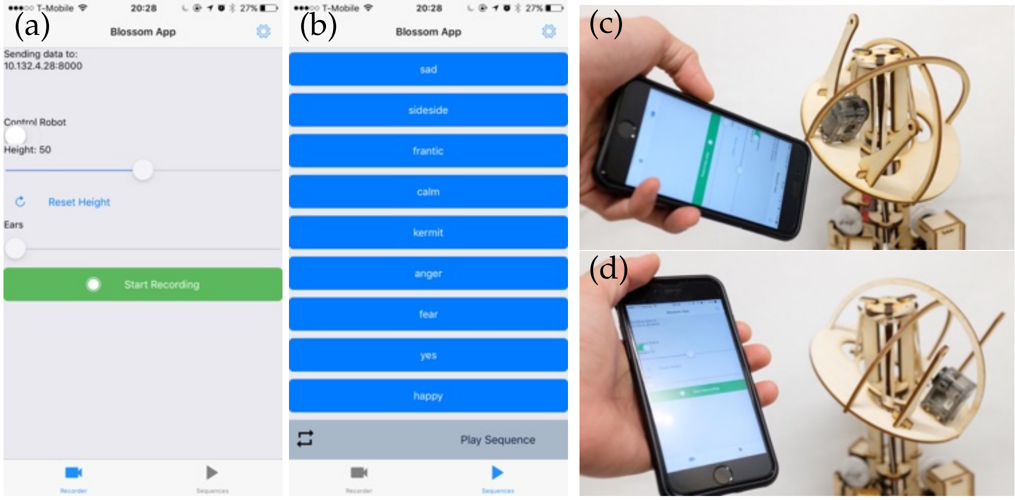


Figure 1.12: Screenshots of the phone app for controlling the robot (a) and playing back gestures from within the app (b). The orientation of the phone is mapped to the orientation of the robot's head (c and d).

surement unit (IMU) to map the phone's orientation to the orientation of the platform. Phone data (kinematic orientation, slider positions) is sent from the phone to the robot using the same RESTful API as previously mentioned. The inverse kinematics of the robot as derived in Appendix 1.5 is used to determine the motor positions required to achieve a given orientation. Currently, the IMU only controls the 3D orientation of the head, but not the vertical offset of the platform's height. This is because integrating the IMU's raw accelerometer measurements at the current data rate (approximately 10 Hz) would quickly result in sensor drift. To solve this, a slider adjusts the platform's resting height. Another slider controls the appendage motor. A mirror mode can be toggled to reflect the motion horizontally to make it easier to control the robot while it faces the user. Gestures can be recorded and played back within the application and can also be looped indefinitely to make idling motions such as breathing or looking around.

1.6 Appearance

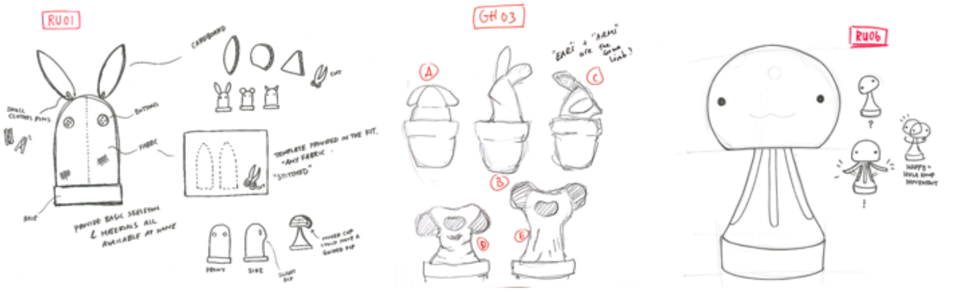


Figure 1.13: Concept sketches exploring different embodiments and movements. The sketches show ideas for interchangeable exterior shapes, and appendages, meant to be hand-crafted by end-users.

The robot's flexibility extends to its outer appearance design. Its exterior is created from soft fabrics that are not rigidly attached to the interior skeleton, and its appendages are interchangeable and in principle open to any tensile mechanism. Concept sketches from the ideation process of various exterior options are shown in Figure 1.13, illustrating the flexibility in the robot's appearance.

1.6.1 Soft Exterior

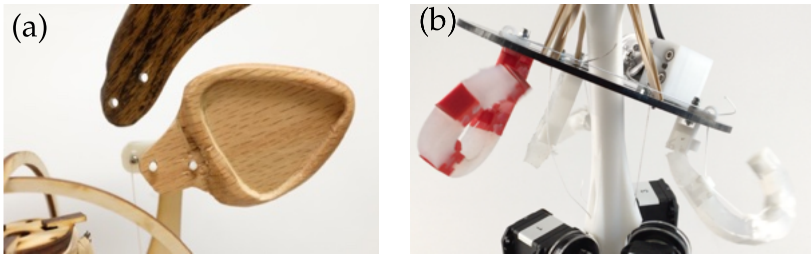


Figure 1.14: Two examples of the swappable appendages: (a) two versions of pluggable wooden ears and (b) flexible silicon arms. Both appendages are actuated using the same tensile mechanism from the appendage motor mounted on the main head platform.

The soft woven exterior of the robot supports expressiveness in two ways: by augmenting the compliance of the internal mechanism through its bending and folding, and due to the fact that it flows freely over the structure. This helps the robot to appear more lifelike by accentuating the organic movement and providing mechanical flexibility to its exterior. Using traditional crafts rather than CAD and rigid manufacturing techniques also supports the design goal of accessibility by enabling a diverse user population to participate in robot-building.

Three examples of crocheted covers are shown in Figure 1.1, one in the likeness of a blue bunny clown, one in the shape of a gray mouse or cat, and the third modeled after a blue jellyfish. They are knit or crocheted out of wool. The blue-and-white design is constructed as a single pull-over piece; the exterior for the mouse design is also single piece but it is open at the top and closes with a button in the back of the head; the jellyfish cover is made of two pieces (one for the head and one for the lower body) that button together at the base of the head. The covers are designed to be loose-fitting to support the organic movement aesthetic and to not constrain the actuation mechanism.

1.6.2 Swappable Appendages

The robot's flexibility is further emphasized by its swappable and open-ended appendage mechanism. The head platform features an additional motor that can interface with various accessories and appendages matched to different exterior designs. Control of the appendages is also tensile, with the motor reeling in a cable and either gravity or an elastic element restoring the DoF.

Figure 1.14(a) shows the mechanism for the ears. The ears attach to posts on a rotating hinge adapter with two hooks, allowing them to be easily interchanged. The hinge adapter itself is tethered to the accessory motor. The jellyfish configuration features flexible arms as shown in Figure 1.14(b). The arms are fabricated by first 3D printing a "skeleton" mold which is then filled with silicone[2]. The rigid skeleton segments act as vertebrae with the silicone serving as ligaments that connect all of the segments. In both cases, gravity restores the DoF. We implemented two examples of swappable appendages, but in theory any single tensile DoF could be added to the robot's design. In the prototyping phase of the robot's design, we explored tails and spinal spikes as additional DoFs.

1.7 Case Studies

To evaluate the extent to which Blossom achieves its design objectives, we have deployed it in the field in four contexts. These deployments were useful in getting feedback on the design and provided insight on how Blossom can interact with a diverse range of users.

1.7.1 Providing the Design to External Research Groups

We provided Blossom prototypes to several external research groups. These collaborations have been useful in evaluating Blossom's accessibility as a research platform by testing the reproducibility of the design.

The first prototype was sent completely pre-built to a company-based re-

[2] The silicone used is Smooth-On EcoFlex™50.

search team studying robot companions for children with autism. The research team was able to set up the robot and control it from the RPi. They then used it in technology demonstrations when meeting with therapists and user populations. A second prototype of the robot was given to a university-based research team. We provided the basic components (as laid out in Figure 1.7) and a repository with the assembly instructions and software library [139]. The group was able to successfully build the robot, install the software, and enlist the help of volunteers to crochet new covers. The group has since implemented the robot in their own field studies. A third prototype was assembled by another university research group. Unlike the previous groups, we provided no components and gave only a link to the repository containing the laser cutting design files, software, and instructions [139]. Apart from troubleshooting some software-related issues, the research team was able to independently build and control the robot.

The gradual open-sourcing of Blossom, from shipping a completely assembled robot to only linking to a design repository, has supported the open-sourcing of the robot and provided growing evidence for the accessibility of the presented design and its potential to be used by a variety of users. The fact that external research groups were able to build the robot with little assistance and readily use it for their own research work has shown that the robot is easily reproducible and that an accessible open-source platform could be a useful model for social robotics research.

1.7.2 Public Exhibitions

Blossom has been exhibited at several public events, including two technology fairs, an academic conference, and a collegiate project team showcase. These events had diverse demographics of attendees, from lay-users to roboticists, and were opportunities to present Blossom to a wider audience and receive feedback on its design. During these events we showcased Blossom's movement and customizability and explained the motivation for the project. Participants responded positively to the robot's design, and several indicated that they would want to interact with it like a pet. At the project showcase, we showed different configurations and allowed participants to control Blossom with the phone. Though many found the controller to be somewhat difficult at the beginning, they found the interaction to be entertaining and would use Blossom to gesture to their friends, supporting to the robot's expressive capabilities.

Along positive comments regarding the design, there were a few recurring questions and suggestions. A common question was whether Blossom could react to user input and whether it had sensors such as cameras or microphones. Attendees familiar with fabric-making expressed interest in creating covers and accessories and sharing the project with a broader craft-making community. Others suggested interfacing Blossom with voice-based assistants to provide them with a physical embodiment. Many also expressed interest in owning or building a Blossom robot.

Showcasing Blossom at these events was useful in demonstrating its expressiveness and receiving feedback from a diverse population of users. The largely positive comments regarding Blossom's appearance are encouraging and affirm that the design appeals to a wide audience. The difficulty that participants had with controlling the robot suggests that there is a learning curve to using the phone as a controller. That said, the ability for untrained users to use a phone to readily create gestures appeared to be more accessible than using traditional programmatic methods.

1.7.3 Children's Science Day

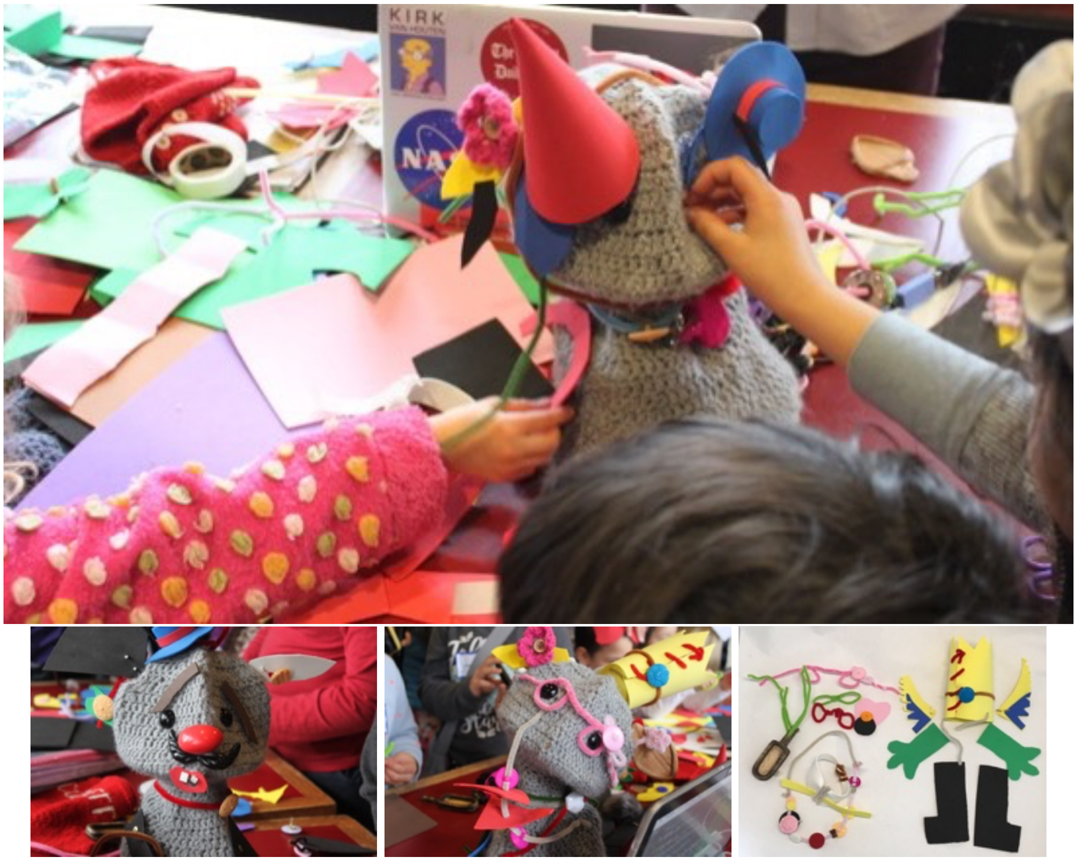


Figure 1.15: Children interacting with Blossom at the science day event (top) and examples of accessories created by participants (bottom).

Blossom was exhibited at a children's science day event where young children, approximately 4-8 years of age, could visit stations with various activities (Figure 1.15). For our activity, we had craft materials available for children to create accessories for Blossom. Children would then affix the accessories to Blossom and control the robot using a smartphone. Participants interacted with Blossom in different ways, with some staying at the booth for a long time crafting several accessories with others only interested in controlling the robot. There were some children who came in groups and took turns between crafting accessories and controlling; these groups sometimes collaborated by having the crafter ask the controller to move the robot to make it easier to attach an accessory. This might suggest that the Blossom platform can encourage collaborative design and interaction of several users with a single robot.

Although we initially suggested creating ears, we were positively surprised that children branched off to make a wide range of different accessories, from appendages to facial features to jewelry. Most creations were simple single-layer shapes, but some designs were more elaborate and featured multiple layers and adornments. The diversity of accessories made emphasizes the flexible design of the platform.

The ways that children controlled Blossom led to interesting observations regarding the smartphone as a controller. Users would often move the phone in exaggerated ways that Blossom would physically not be capable of achieving, such as turning completely upside down or twisting around over 360°. The children also had their own implicit feature requests, such as how to make Blossom locomote and jump. These were emphasized by that fact that several children chose to make appendages such as legs and wings.

Adults were also interested in Blossom, from the project's application to its technical implementation. Some parents participated by making their own accessories while others helped their children control the robot more effectively.

They commented on the project as relating art and technology, fitting in with Science, Technology, Engineering, Arts, and Math (STEAM) education, and also noted Blossom's unconventional appearance compared to traditional robot aesthetics.

The children's science day was a valuable opportunity to demonstrate Blossom's aesthetic flexibility and the accessibility of its customization and control method, even to very young users. The positive responses to the activity from children and adults alike show that they enjoyed the interaction and further supports the platform's expressiveness. We especially noted that the flexible design of the robot supported users with diverging interests.

1.7.4 Build-a-Blossom Workshop

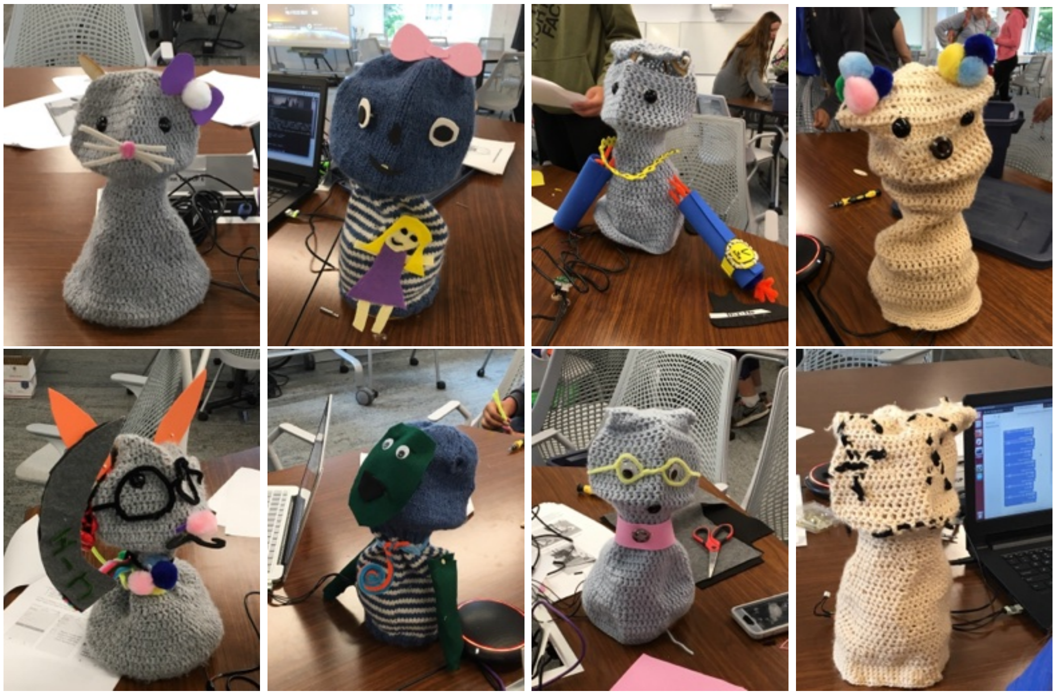


Figure 1.16: Examples of the embodiments created by the students in the building workshop.

Lastly, Blossom was used in educational workshops for middle school students to learn about the skills involved in robotics engineering. The students had varying levels of technical experience, ranging from good familiarity with technology to very little exposure to programming or mechanical construction. The activity was to build and customize a Blossom robot, program its gestures, and choreograph its movements to a video of the students' choosing. There were six workshop sessions; each was approximately 80 minutes long and had 16–20 students that were divided into four groups. The total was 107 students in 24 groups. Lab members familiar with the construction and programming processes were present to provide assistance, but intervention was kept to a minimum and mainly involved guided troubleshooting.

Each group was provided a partially-disassembled robot and the assembly instructions. The construction process included building the head platform and attaching the ears, connecting the tower to the base assembly, connecting the motors, and suspending the head by hanging it from the tower and attaching the cables from the motors. A crocheted cover was included with each robot. We observed that often some students were building the inner structure, while other group members customized the cover with craft accessories. Figure 1.16 shows examples of some of the appearances created by workshop participants.

Once the robot was assembled, students connected it to a computer and programmed its movements using the smartphone application. Often groups designated one member with the mobile application to be the movement choreographer in charge of creating gestures. Students then imported the gestures into the web application and timed each movement, some with modulation, to the video chosen by the group. This resulted in a variety of choreographies with which the robot reacted to the student's videos. The videos themselves ranged from music videos to which the robot was made to dance, often "dressed up" as the performing artist, to humorous videos with the robot reacting as an audience. Other examples included viral videos ("memes") where the robot was fashioned like one of the characters in the video, imitating the action on screen.

All of the 24 groups were able to successfully build and control the robot by the end of their session. The structures were mostly assembled correctly, except for the ear assembly, which had sometimes to be bypassed due to its cable routing. The programming process was largely error-free and some groups were able to make fairly complex choreographies. Similar to our observation at the children's science day event, many students tried to control the robot in impossible manners.

The vast majority of students were actively engaged throughout the sessions. We conducted brief informal question-and-answer sessions at the end of each meeting, where students were asked to say what their favorite and least favorite part of the workshop was. There was a wide variety of responses about the favorite part, with some students enjoying the craft more, and others preferring the mechanical construction or the gesture generation. This suggests that the Blossom platform allows students with different interest to be involved in some capacity. Others expressed satisfaction at being able to build and control a complete working robot in a short time. Several students who were admittedly disinterested or intimidated by robotics at the beginning found themselves enjoying it due to the engagement of the activity and the relation to personally meaningful video content.

The workshop was an opportunity to thoroughly evaluate all of Blossom's major design objectives. The fact that untrained middle-school students were able to build and animate the robot within the duration of the sessions demonstrated the accessibility of the platform's assembly and gesture authoring workflow. The variety of embodiments and their relation to the personal content choice of the students emphasized Blossom's flexibility. The complexity of the resulting choreographies indicates the robot's expressiveness.

The difficulties in assembly highlighted weak points in the design that can be rectified in future iterations, most notably the appendage module. The interconnectivity between the robot's control computer and the phone controller can also be streamlined. Future evaluations on customizability should explore alternative embodiments by providing different appendages.

1.8 Future Work

The field deployments of Blossom, together with our own experience in manufacturing and using Blossom, indicate several points in which the current design can be improved upon.

Smartphone Control Mapping Many users who attempted to control the robot tried to move it in ways that it was not capable of, such as turning all the way around and flipping. This reveals a problematic mapping between the unconstrained motion of the phone and the limited range of the robot. Possible solutions include better instructions or training to control the robot properly, a mechanical rig to place the phone into, enforcing the robot's movement constraints, or methods for better mapping from the raw orientation detected by the smartphone's IMU sensor to the robot's pose. Relatedly, many users attempted to control the height of the platform by raising and lowering the phone; while it would be difficult to get accurate height control due to the sensor used, usable height control should be explored, possibly by using filtering or predictive methods to alleviate drift.

Sensing Capabilities Users often commented that they wished Blossom had sensing capabilities. Incorporating sensors for the robot to react to external inputs should thus be considered. Implementing sensors on the robot itself may compromise its accessibility and handcrafted aesthetic, but simple sensors could afford richer functionality without being obtrusive. Many have interacted with Blossom by petting its head or calling to it, and components such as touch sensors or microphones could be implemented to provide more functionality. Another approach is to leverage sensors built into smartphones [140], such as the microphone or camera to avoid adding complexity directly to the design of the robot itself.

Intermediate Programming Language The Blockly interface is currently only used for triggering gestures to videos, but it could also be used as a mid-level programming method that is more versatile than the existing Wizard-of-Oz interfaces, while being still more accessible than a full programming language. Features such as motor control and conditional statements responding to external inputs could be useful to expand the current functionality.

Lower Cost On the mechanical side, while wood is relatively inexpensive and is well-aligned with the handcrafted aesthetic of the robot, transitioning to an even cheaper material such as cardboard or paper could further improve its accessibility. The most expensive aspect of the current design are the high-end servo motors. They provide many advantages over standard servo motors, primarily velocity and acceleration control and daisy-chaining, but are relatively expensive. Transitioning to standard hobby servos would significantly reduce the overall cost of the platform at the potential cost of ease-of-control and movement quality.

Diverse Appendages Finally, we would like to explore more kinds of appendages to illustrate the platform's customizability. Flexible arms and dinosaur-like spikes were briefly explored, but the ear design has proven to be the most easy-to-use and expressive. Given the inclusion of limbs and wings among the accessories created at the children's science event, such alternative configurations should be explored in the future. Different appendages may also affect the robot's expressiveness by altering its DoFs and therefore its gesture capabilities.

Part III

Movement

CHAPTER 2

MOVEAE: MODIFYING AFFECTIVE ROBOT MOVEMENTS USING CLASSIFYING VARIATIONAL AUTOENCODERS

2.1 Introduction

In this work, we demonstrate the use of neural networks to modify the affective qualities of movements for an expressive robot. Current robot movement generation methods demand a deep understanding of the domain and its feature space, rendering these processes costly and hard to implement. Conversely, neural networks used in deep learning are able to learn the feature space on their own, reducing the dependency on domain knowledge. Neural networks may thus be applicable to the creation of expressive robot movements.

Robots designed for social interaction are becoming more common in spaces such as homes and storefronts. Movements and gestures are important modes of nonverbal communication that are unique to robots compared to other agents without physical bodies [66, 141]. There are various methods for creating expressive movements, from manual trajectory editing interfaces to learning from demonstration (LfD) [70]. However, these methods can be slow and often require prior knowledge of a specific robot platform. These techniques are thus difficult to implement and lacking in generalizability. To more quickly create new behaviors, roboticists sometimes turn to adjusting affective qualities of existing robot movements [142, 143, 144, 145]. Adjustment is easier than authoring new movements, but still requires technical knowledge of kinematics and movement theory. These pitfalls lead to robot behaviors usually being preprogrammed, creating a novelty effect that stunts long-term interaction and conveys a lack of intelligence [146, 147].

At the same time, advancements in deep learning have enabled the creation of data-driven neural network models that can learn complex features given sufficient data. These have enabled various applications ranging from temporal forecasting to image generation [148]. While these methods have seen success in tasks such as audiovisual perception and generation, they have remained largely unadopted for generating robot behaviors, where most algorithms are based on traditional machine learning methods or rely on problem-specific heuristics [94]. Neural networks can reduce the dependency on domain knowledge and heuristics by learning the features directly from the input data. Recently, neural networks have been developed for modifying high-level features in domains such as images [72] and audio [73] by editing low-level parameters in a learned ”latent” embedding space. These works used the same approach for both images and audio, showing that neural networks can be more domain-agnostic and generalizable than heuristic methods.

To address the problem of repetitive movements in interactive robots, we propose to use deep learning techniques, particularly variational autoencoders (VAEs), classification networks, and latent space editing methods, to modify affective movement features for a low-degree-of-freedom (DoF) robot. We first learn low-dimension latent representations of the robot's affective movements. These latent representations can be used to both reconstruct the original movement and classify the movements by the intended emotion (happy, sad, angry). We then modify the valence and arousal features of the movements by using simple arithmetic operations in the latent embedding space. Our contributions are:

- A classifying variational autoencoder neural network architecture that compresses expressive robot movements into a lower-dimension latent space. The lower-dimension latent representations can reconstruct the original movements and are separated by emotion class.
- A method using linear regression to map the latent space representations into the circumplex emotion model dimensions of valence and arousal.
- An algorithm and interface for modifying the valence and arousal of the movements.
- Objective and subjective evaluations to assess the validity of this approach, in the form of neural network performance metrics and an online survey.

2.2 Related Work

We review works in affective robot movements and neural network applications for affective robotics and latent feature modification.

2.2.1 Affective Robot Movements

Many prior works in human-robot interaction (HRI) categorize robot emotions into discrete classes according to Ekman's six categories: happiness, sadness, anger, surprise, fear, disgust [149]. In contrast, the circumplex model places emotion classes on the continuous dimensions of valence and arousal [150], with valence corresponding to positivity and negativity and arousal corresponding to high and low energy. The circumplex model illustrates the qualitative relationships between the emotions and its continuous dimensions are conducive for quantitative operations, making it suitable for adoption in numerical models.

2.2.2 Robot Movement Generation / Modification

Movements and gestures are primary ways for robots to express their internal emotive states, and methods for designing affective robot movements have been extensively studied [94].

Generation

There are many approaches for generating robot movements, from low-level manual trajectory editing to high-level demonstrative techniques such as LfD [70]. These methods, however, have several drawbacks. Editing trajectories is time-consuming and unintuitive for non-roboticists, while directly manipulating a robot for LfD may be difficult to perform in real-time. LfD can be performed indirectly by attaching sensors to a human demonstrator, but this introduces the correspondence problem of mapping a human movement to a non-human embodiment. This has been addressed in many works within the graphics community, often using heuristic mappings from human poses to animate animals or other creatures [151, 152, 153]. Alissandrakis et al. explored heuristic methods to address this correspondence problem for robots [154], though their approach required extensive knowledge of the embodiment's kinematics. These difficulties lead robot movements to be largely preprogrammed and repetitive.

Modification

Modifying existing movements can be used to quickly expand a robot's library of movements, but still demands a high level of technical knowledge. As discussed by Karg et al. [94], most techniques used to modify affective robot movements rely on prior heuristic knowledge of robotics and the kinematics of a specific platform. These approaches typically adjust movement features that have been empirically found to be important for conveying affect, such as gaze direction [142, 143] or speed [144]. Desai et al. used a simulation of a quadrupedal robot with editable movement parameters such as walking pattern, speed, and body angle to adjust the affective quality of its gait [145]. The interface and method used was accessible compared to manual trajectory editing techniques, but still required a high level of domain knowledge.

2.2.3 Neural Network Applications for Affective Robotics and Latent Feature Modification

The strength of neural networks compared to heuristic methods is their ability to learn complex and intractable data features with less dependence on domain knowledge and manual feature engineering. Neural networks have found success in complex applications for affective computing, primarily in perceptual tasks such as emotion recognition [155, 156], though some works have explored using neural networks for affective speech and expression generation [157, 158].

Affective robotics

Apart from emotion recognition, there have been few applications of neural networks for affective robotics. With regards to movement generation, Rodriguez et al. used a generative adversarial network (GAN) to generate talking gestures for a Pepper robot [159], but mostly generated random movements that did not consider affect. In more affect-oriented work, Zhou et al. compared hand-designed and network-learned feature costs for editing affective handovers [160]. The results showed that the hand-designed features were more suitable for expressing simple styles such as happy and sad, but the network could be preferable for complicated styles such as hesitant. This suggests that neural networks may be a better option for more complex affect expression.

Latent feature modification

Autoencoders are neural networks that learn a latent space to compress high-dimensional data into low-dimensional representations. The learned latent space can also be used to modify high-level features by editing the low-level parameters. Larsen et al. used this approach to modify discrete features of face images, such as gender and facial hair [72]. Roberts et al. extended this technique to modify continuous features of music, such as note density and pitch [73]. These works used the same general techniques for two very different domains, demonstrating the potential to use neural networks for modifying data features with less domain knowledge compared to heuristic methods.

The capabilities of neural networks for feature modification can be applied to affective robot movements. This intersection of HRI and deep learning can mitigate the novelty effect by continuously updating a robot's behavior library. An ever-growing repertoire of behaviors would help imbue robots with a sense of affective autonomy and may promote prolonged human-robot interactions.

2.3 Neural Network Background

Neural networks are the foundational models used in deep learning, approximating a transfer function from input data to output predictions. Compared to simple linear perceptrons [161], modern neural networks use varied activation functions, convolutions, and recurrence in their layers to create a non-linear model between the input and output. These layers can be arranged into various network components such as encoders, decoders, or classifiers. Network components can then be combined into larger architectures such as image classifiers [162], recurrent networks [163], and autoencoders for dimensionality reduction [164]. Neural networks are trained by defining loss functions for the desired objectives, such as categorical cross-entropy for classification or mean error for reconstruction. Before training, the input data is split into training and testing sets. The training set is repeatedly passed through the network to optimize the layer parameters to minimize the loss functions and achieve the objectives. The test set is held out and does not update the network parameters, but is instead used to validate the model's performance on unseen data.

2.3.1 Variational Autoencoders (VAE)

The primary network architecture used for this work is a variational autoencoder (VAE), which compresses input data into a latent embedding space while also giving this space a known structure.

Autoencoders are comprised of two components: encoders to compress the input data into a latent space, and decoders to decompress the latent space into reconstructions of the original inputs. Traditional autoencoders seek to minimize the reconstruction loss, which is defined as the difference between the input data and output reconstruction. VAEs additionally implement a Kullback-Leibler (KL) divergence objective, which structures the latent space into a Gaussian distribution. β-VAE is a further modification that implements weighing between the reconstruction and KL loss, allowing for the relative importance of the objectives to be tuned [165].

The combination of the reconstruction loss and KL divergence ensures that decoding from a random sample in the known latent distribution results in a valid realistic data sample. In lieu of random sampling, the original data can also be edited in the latent space to modify high-level features. This has enabled the use of VAEs in various applications such as image modification [72] and musical style transfer [73]. GANs were also considered and can extend VAEs to achieve better results [72], but their notorious training difficulty makes simple VAEs a better choice for our purpose.

2.3.2 Latent Space Editing to Modify Features

Latent space modification in the aforementioned prior works [72, 73] was achieved by calculating "attribute vectors" $\vec{a}_f$ in the latent space for modifying high-level features f (e.g., hair color, musical pitch). The vector $\vec{a}_f$ can be seen as a latent-space translation in the direction of data points that contain the feature of interest.

Given a latent-space representation of a data sample $\vec{x}_0$, the high-level features are modified by adding these attribute vectors. The degree of modification for a given feature is controlled with a weight parameter c_f.

$$\vec{x} = \vec{x}_0 + \sum_f c_f \vec{a}_f$$

The modified latent representation $\vec{x}$ is then passed through the decoder of the VAE to generate the new modified data sample.

In the face image modification work mentioned above [72], the features were binary (e.g. mustache or no mustache, blonde or not blonde). The attribute vectors were calculated as the difference between the mean latent vectors $\vec{\mu}_f$ of the "yes" and "no" groups.

$$\vec{a}_f = \vec{\mu}_{f,yes} - \vec{\mu}_{f,no}$$

For music modification [73], features were continuous (e.g. note density, pitch, average interval). The attribute vectors were calculated by first ranking the samples in terms of intensity (e.g. high vs low note density) and taking the difference between the mean latent vectors of the highest and lowest quartiles.

$$\vec{a}_f = \vec{\mu}_{f,Q_{high}} - \vec{\mu}_{f,Q_{low}}$$

2.4 Implementation

To illustrate our approach, we implemented a system to generate gestures expressing three emotions on a small desktop robot.

2.4.1 Robot Platform

We used the Blossom robot, an open-source social robot (Figure 2.1) [166]. Blossom's internal mechanisms consist of a head platform suspended from a tower structure that rotates about its base platform. Blossom features four degrees of freedom (DoFs): roll, pitch, yaw, and vertical translation, though we disable vertical translation to simplify the control interface. The robot achieves motion with four actuators: tower motors 1, 2, and 3 control the front, left, and right sides of the head, respectively, and a motor in the base rotates the tower left and right. The robot's head can pitch up and down and roll left and right ±45°and yaw left and right ±150°about its base. Although the robot's DoFs are limited compared to more complex embodiments, it features a large range of motion and head movements alone can convey complex affective information [167]. Users can control the robot with a mobile application that maps the orientation of the phone into motion for the robot's body.

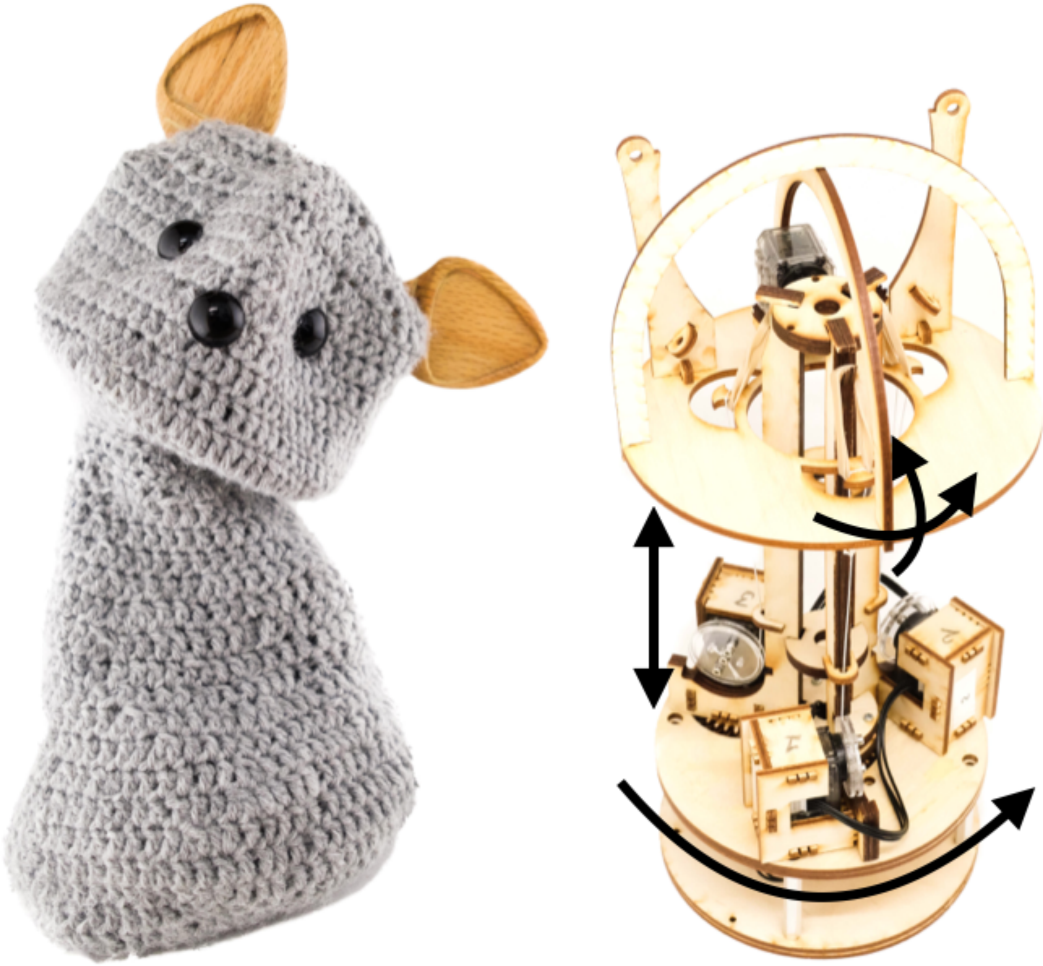


Figure 2.1: The Blossom robot. The exterior (left) is made of soft materials while the interior mechanism (right) consists of a central tower structure from which the head platform is suspended by elastic bands. The head platform has four degrees of freedom: roll, pitch, yaw, and vertical translation.

We collected a dataset of emotive Blossom movements by asking volunteers to puppeteer the robot to display three main emotions: happy, sad, and angry. Movements are created with a phone application that translates the movement of the phone directly into the movement of Blossom's head. The dataset consists of approximately 25 movements per emotion class, each recorded at 10 Hz. Because neural networks require the input data to be consistently-sized, the movements are cut down by chunking them into sliding three-second windows every 1.5 seconds (Figure 2.2). The resulting dataset thus contains over

5,000 120D samples[1].

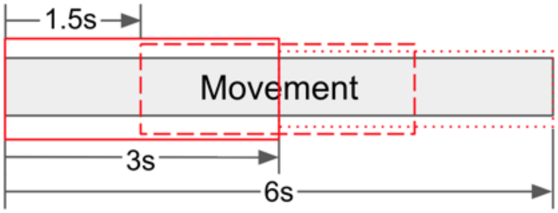


Figure 2.2: Illustration of how the movement data is ”chunked” into three-second windows with 1.5-second overlaps to be used by the network. In this example, this six-second movement will yield three samples.

2.4.2 Neural Network

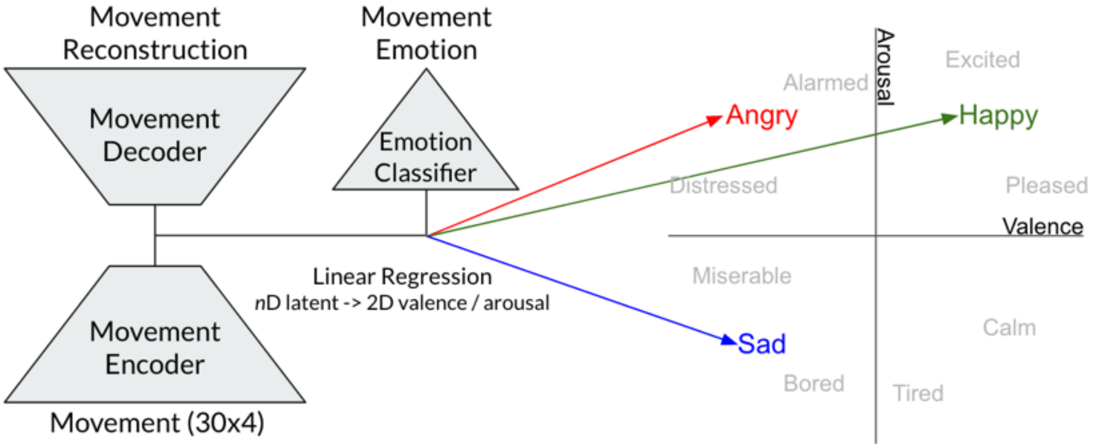


Figure 2.3: The network architecture consists of a variational autoencoder (left) with an emotion classifier (center). Once the network is sufficiently trained to reconstruct the movements and classify the latent representations by emotion class, linear regression is used to map the *n*D latent space into the 2D circumplex model (right) with the valence and arousal dimensions.

We constructed the neural network with Keras and TensorFlow [168].

[1] 30 (3 seconds, 10 Hz) points × 4 DoF = 120D.

Classifying VAE Architecture

Figure 2.3 shows the network architecture which consists of a VAE with an additional emotion classifier. The role of the VAE is to compress the 120-dimensional input movement data into a lower-dimension latent space. The classifier ensures that this latent space is separable by the emotion classes (happy, sad, angry). The network is based on convolutional layers and the parameters are detailed in Table 2.1. We chose convolutions over recurrence due to easier training and adjustable temporal reception [169]. The number of filters corresponds to the number of kinematic features to detect. Kernel size controls the receptive field, with a larger size denoting increased temporal correspondence between timesteps. Dropout was used to reduce overfitting given the small data size. The training objectives are:

- Reconstruction loss to ensure that the output reconstructions are identical to the input data.
- KL divergence to give the latent space a Gaussian structure.
- Classification loss to separate the learned latent space by emotion class.

Latent Space → Circumplex Model

The dimensions in the learned latent space do not meaningfully represent human-readable affect. In order to both visualize the gestures and allow users to modify them, we use linear regression to map the latent space onto the circumplex model's valence and arousal dimensions. First, we calculate the centroids of each emotion class in the *n*D latent space. Each centroid is then recalculated by weighing each sample by the inverse of its distance to the original

Table 2.1: Network Layers and Parameters

	Layer	Parameters
Encoder	**Input**	**Movement (30x4)**
	Dropout	10%
	Conv1D+BN	7 filters, kernel size 5
	Leaky ReLU	$\alpha = 0.01$
	Dropout	5%
	Conv1D+BN	4 filters, kernel size 3
	Leaky ReLU	$\alpha = 0.01$
	Flatten	–
	Dropout	5%
	KL Resample	–
Decoder	Dense	60
	Upsample1D	2
	BatchNorm	–
	Leaky ReLU	$\alpha = 0.01$
	Conv1D+BN	4 filters, kernel size 3
	Leaky ReLU	$\alpha = 0.01$
	Conv1D+BN	6 filters, kernel size 5
	Leaky ReLU	$\alpha = 0.01$
	Conv1D+BN	6 filters, kernel size 5
	Leaky ReLU	$\alpha = 0.01$
	Dense	30
	Output	**Movement (30x4)**
Classifier	Dropout	5%
	Dense	13
	Leaky ReLU+BN	$\alpha = 0.01$
	Dropout	5%
	Dense	3
	SoftMax	–
	Output	**Emotion**

unweighted centroid. We use these weighted centroids to diminish the importance of movement samples that may be confused with another emotion class. An ordinary least squares linear regression model fits the nD centroids of each emotion to their locations on the 2D circumplex model. The circumplex model does not numerically define the emotion locations, so they were arbitrarily chosen as:

- Happy: valence = 1, arousal = 1
- Sad: valence = -1, arousal = -1
- Angry: valence = -1, arousal = 1

After fitting the centroids to their locations, the linear regression model is used to transform all movements into the circumplex space.

Latent Feature Modification

We use a similar approach to feature modification as prior works (see: Section 2.3.2). First, the circumplex representations of the data samples are ranked from high to low intensity for both valence and arousal features. For each feature f, the latent space means for the higher and lower halves are calculated as $\vec{\mu}_{f,high}$ and $\vec{\mu}_{f,low}$ Compared to the quartiles used in prior work [73], splitting into halves was empirically found to yield better performance. A feature's attribute vector $\vec{a}_f$ is calculated as the difference between its high and low mean vectors.

$$\vec{a}_f = \vec{\mu}_{f,high} - \vec{\mu}_{f,low}$$

To modify the valence and arousal of a movement, its original latent representation $\vec{m}_0$ is summed with a linear combination of the attribute vectors and the feature weights c_f.

$$\vec{m} = \vec{m}_0 + \sum_{f=\{V,A\}} c_f \vec{a}_f$$

Modification Interface

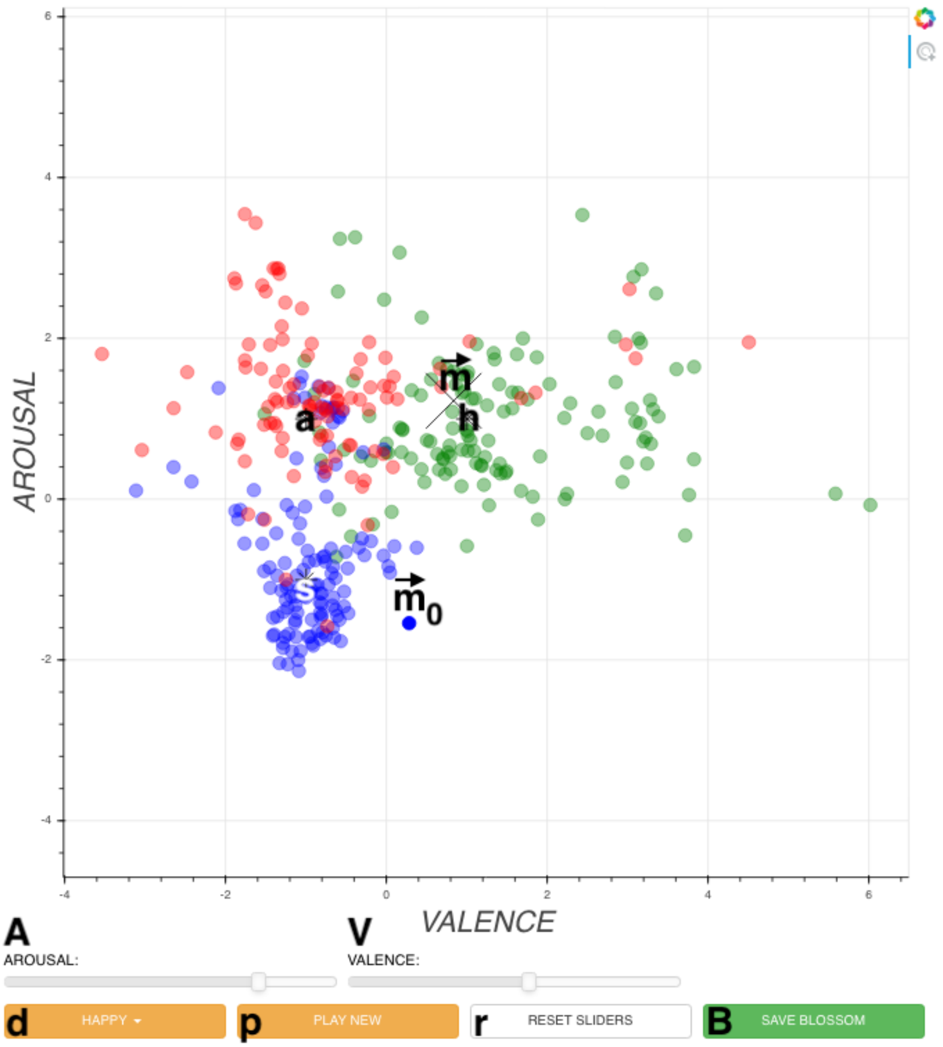


Figure 2.4: Movement modification interface. The emotions are separated by color and their centroids are marked: (*h*)appy is green, (*s*)ad is blue, and (*a*)ngry is red. The selected movement $\vec{m}_0$ is modified by either adjusting the (V)alence and (A)rousal sliders or by selecting an emotion from the (d)ropdown menu, and $\vec{m}$ denotes the location of the modified movement. In this case, the dropdown menu was used to modify a sad movement to be happy, and the sliders updated accordingly. The (p)lay button plays $\vec{m}$ on Blossom, the (r)eset button resets the sliders, and the (B)lossom button saves $\vec{m}$ to a file for later use.

We created an interface for visualizing the circumplex model and modifying the movements (Figure 2.4). Each point on the scatter plot represents a three-second movement sample projected from the latent space into the circumplex model using the regression parameters described in Section 2.4.2. The emotion classes are color-coded and the projected centroids are marked. In the graph, green is happy (h), blue is sad (s), and red is angry (a). The user selects a movement $\vec{m}_0$ and adjusts the attribute vector weights c_f using the valence (V) and arousal (A) sliders. The projected modified movement $\vec{m}$, denoted by the large X marker, updates in real time. In addition to directly adjusting the feature weight sliders, users can also use a dropdown emotion selector (d) to update the attribute vector weights based on the emotion centroids. The dropdown selector uses the valence-arousal distance between the target emotion's centroid and the original movement $\vec{m}_0$ to indirectly update the sliders and c_f. In Figure 2.4, a sad movement at [0.4,-1.6] was modified to be happy, whose centroid lies at [1,1]. Selecting "happy" from the dropdown thus sets the valence and arousal sliders to 0.6 and 2.6, respectively, and updates the movement $\vec{m}$ close to the target centroid. Once modified, the VAE decoder generates a three-second gesture in the form of motor trajectories. The interface also includes buttons to play the movements on Blossom (p), save the modified Blossom movement to a file (B), and reset the sliders (r).

2.5 Evaluation

We evaluate the performance of the neural network and the modification method using objective metrics for each of our training objective, as well as using an online user survey.

2.5.1 Network Parameters for Evaluation

We empirically derived most of the network parameters. The test set hold-out rate was set to 20%. The size of the nD latent space was derived empirically. $n = 40$ was found to be the maximum possible reduction while still achieving the training objectives. For the movement reconstruction objective, using simple mean-squared or mean-absolute error functions resulted in a lack of base motion (yawing) and side-to-side movement (rolling). This may have been due to augmenting the data by mirroring the left-right motions, causing the network to ignore these DoFs and simply default to looking straight ahead. To overcome this issue, we used a custom loss function that weighs each movement DoF differently and uses squared error for the front and base motor and absolute error for the left and right motors. The weights for the front, left and right, and base motors were empirically set to 5, 7, and 20. The KL divergence loss was implemented according to β-VAE [165], and the classifier used categorical cross-entropy as its loss function. During network training, we monitored the following objectives:

- Reconstruction - Monitor loss and plot comparisons of original and reconstructed movements for visual inspection.
- KL - Not monitored, but β-VAE recommends adjusting the weight according to the task [165].
- Classification - Monitor accuracy and plot latent embeddings in TensorFlow Projector to visually inspect emotion class separation in the latent space [170].

We tuned the loss weights iteratively by increasing weights for underperform-

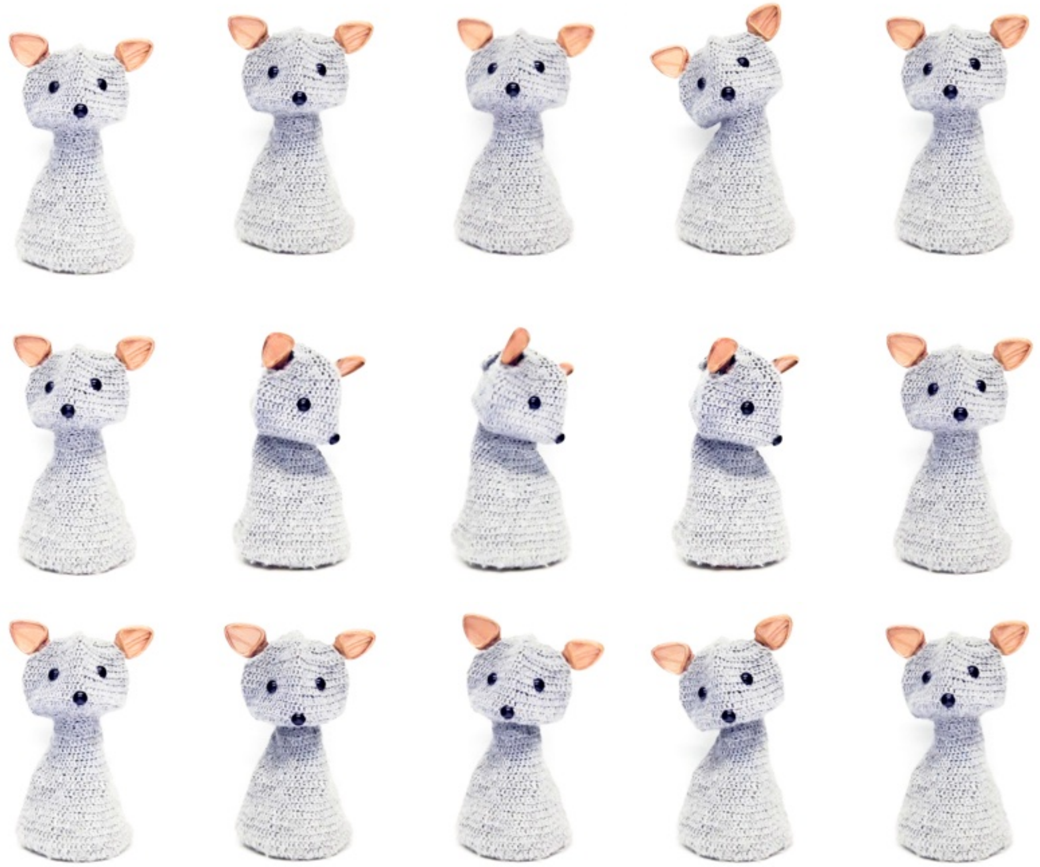
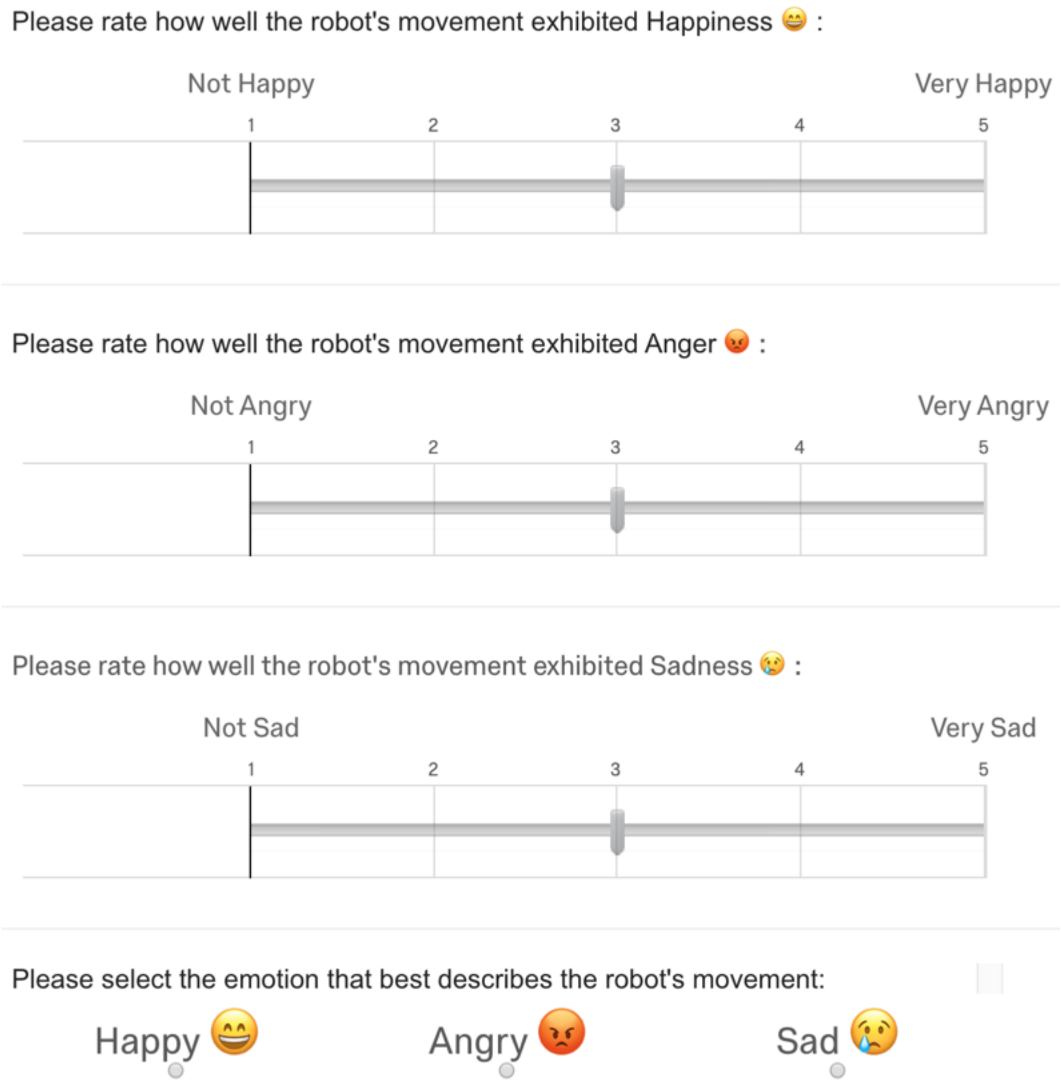


Figure 2.5: Filmstrips of a happy movement (top) modified into sad (middle) and angry (bottom).

ing objectives, e.g. increasing the reconstruction weight if the movement characteristics are not being preserved or increasing the classification weight if the emotions are being confused. We settled on 5, 0.1, and 7 for the reconstruction, KL, and classification loss weights, respectively. We empirically tuned the remaining training parameters: learning rate of 0.1, batch size of 30, Adam optimizer [171], and mixup with a factor of 0.2 [172]. 100 epochs was sufficient to stabilize the losses.



Please rate how well the robot's movement exhibited Happiness 😄 :

Please rate how well the robot's movement exhibited Anger 😡 :

Please rate how well the robot's movement exhibited Sadness 😢 :

Please select the emotion that best describes the robot's movement:

Happy 😄 ○ Angry 😡 ○ Sad 😢 ○

Figure 2.6: Online survey questions.

2.5.2 Online survey

We evaluated the subjective effectiveness of our method using an online survey, which presented videos of gestures along with a questionnaire for each gesture. The movements shown in the online user survey were chosen by randomly selecting five samples within the held-out test sets of the three emotion classes, resulting in a dataset of 15 original movements. We then modify each movement into the two other emotion classes by using the dropdown interface described above, e.g. a happy sample was modified into both sad and angry, as in Figure 3.7. This provides two modified movements for each original movement, resulting in a survey dataset of 45 movements, 15 original and 30 reconstructed.

We had two main hypotheses. If the latent representation of a movement is modified to lie in another target emotion space on the circumplex model, then the modified movement's new emotion:

H1) is consistently recognized as the target emotion.

H2) expresses the target emotion as legibly as an original movement with the same emotion.

For each survey question, a video of a movement was followed by Likert scales for how well it represented each emotion class and a multiple choice selection for which emotion it best represented (Figure 2.6). Each survey showed 30 random movements from the original 45. We distributed the survey using Amazon Mechanical Turk offering $2 compensation and received 100 responses.

2.6 Results

The performance of this approach was evaluated using both objective metrics for the technical implementation and statistical significance tests for the survey results.

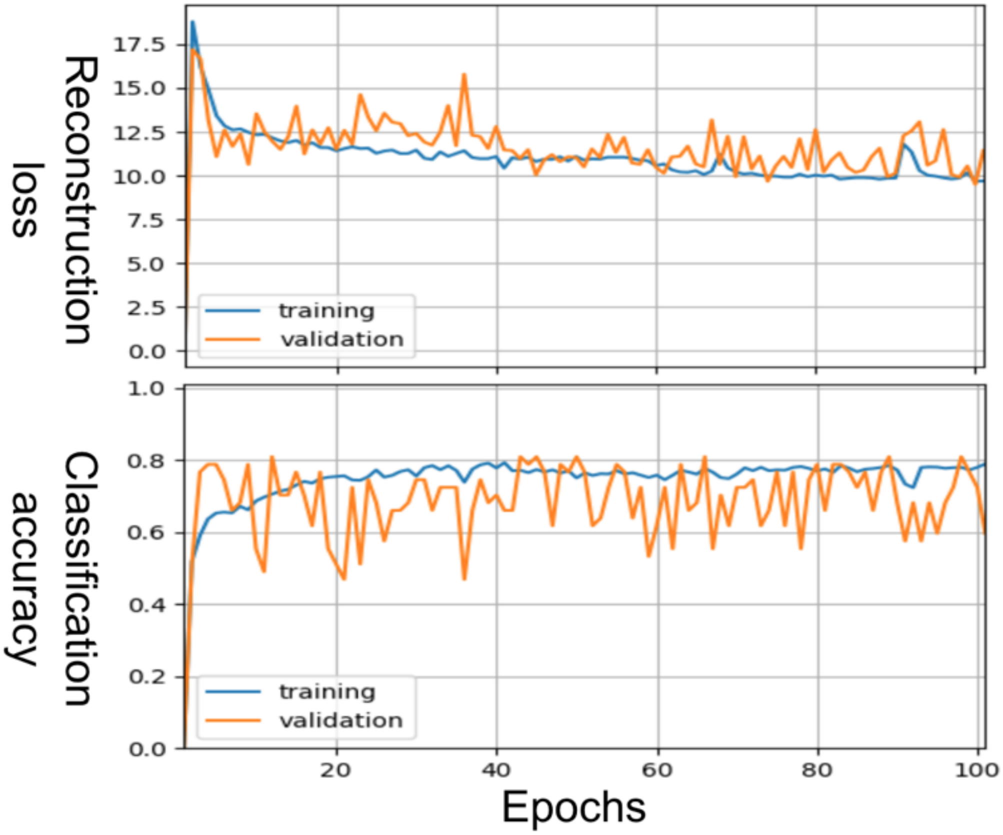


Figure 2.7: Movement reconstruction loss (top) and emotion classification accuracy (bottom) over 100 epochs.

2.6.1 Objective Metrics of Network Performance

We used traditional neural network training metrics to objectively evaluate the technical implementation. The movement reconstruction loss and emotion classification accuracy are the primary training objectives. The KL divergence was weighed lowly as it is comparatively unimportant and primarily provides the Gaussian structure for the latent space.

Figure 2.7 shows the training curves for the movement reconstruction loss and emotion classification accuracy. Both curves leveled off by the end of training. The validation curves, while noisy, are very close in performance to the training curves, suggesting that the model did not overfit to the training set. We

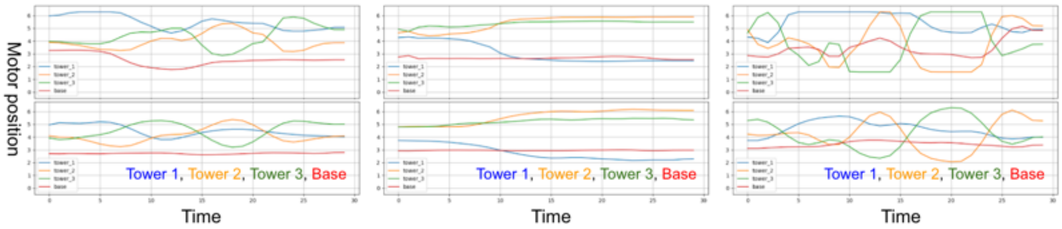


Figure 2.8: DoF curves for original (top) and reconstructed (bottom) movements for each emotion (happy left, sad center, angry right). The blue, yellow, green, and red lines represent the front, right, left, and base motors, respectively. The reconstructions have difficulty achieving the same exaggeration as the original movements, but retain the overall trajectory characteristics.

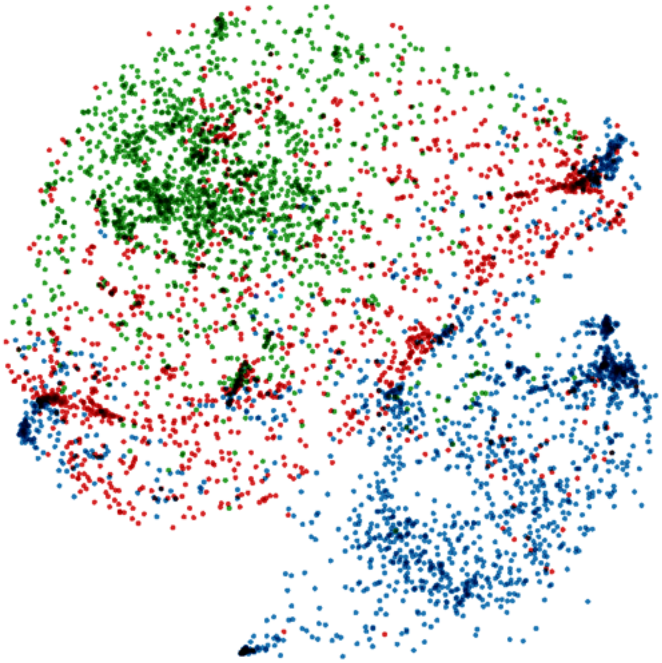


Figure 2.9: t-SNE representation of all of the movement samples in the latent space. The latent space is visibly separated by emotions (happy is green, sad is blue, angry is red).

achieve close to 80% classification accuracy, which is promising considering the abstract nature of the movement data and simplicity of the network.

The reconstruction objective is further evaluated by comparing the original and reconstructed movements. Figure 2.8 contrasts original (top) and reconstructed (bottom) samples for each emotion class. The reconstructions are less exaggerated, but capture the overall trajectory characteristics of the original movements.

The classification objective is further evaluated with visualization of the latent space. Figure 2.9 is a dimensionality reduction of all movement samples in the latent space using t-SNE [173]. The emotion regions are visibly separated in this space even before applying the transformation into the circumplex space.

Feature Sliders

The performance of the feature sliders can also be objectively measured. A slider would ideally modify a movement along only its intended feature axis (e.g. the valence slider moves a movement sample only along the horizontal valence axis in the interface). However, editing in the latent space may induce coupling in the features, i.e. modifying valence may indirectly modify arousal, and vice-versa. This coupling was also present in prior work [72], where adding mustaches also added masculine features due to these features being highly correlated in the input dataset.

The degree of feature coupling is highly dependent upon the emotion class and specific movement sample. To test this, each slider was maximized individually and the unit difference vector $\widehat{m}_\Delta$ from the original $\vec{m}_0$ to modified $\vec{m}$ movement was calculated.

$$\widehat{m}_\Delta = \frac{\vec{m} - \vec{m}_0}{|\vec{m} - \vec{m}_0|}$$

The dot product between $\widehat{m}_\Delta$ and the unit feature vector (¡1,0¿ for valence,

¡0,1¿ for arousal) denotes the alignment of the modification direction and the intended axis, with a dot product of 1 denoting perfect alignment. This was calculated for every movement in the held-out test set, and the mean dot products for all emotion-feature combinations are presented in Table 2.2. All of the results are almost 1, indicating that both sliders move primarily in their respective axes and perform as intended.

Table 2.2: Slider evaluation results.

		Feature	
		Valence	Arousal
Emotion	Happy	0.999	0.996
	Sad	0.995	0.989
	Angry	0.995	0.992

Dropdown

The performance of the dropdown menu for modifying a movement towards a target emotion can also be objectively measured. The dropdown emotion selector indirectly adjusts the sliders by setting the valence-arousal distance from the movement to the target emotion's centroid as the slider values. As visualized on Figure 2.4, the effectiveness of this method can be calculated by measuring the distance between the final modified movement $\vec{m}$ and the target emotion centroid (h in this example), with a distance of 0 denoting ideal performance. This distance was calculated for every movement in the held-out test set, and the mean distances for each original-target emotion combination are presented in Table 2.3.

Modifying a movement towards its original emotion yields the best performance. For cross-emotion modification, sad→happy performs the best, followed by angry→happy and happy→sad. Interestingly, happy and sad both

Table 2.3: Dropdown evaluation results. Bolded values indicate best performance for each original emotion class. Italicized values indicate second-best performance.

		Target emotion		
		Happy	Sad	Angry
Original emotion	Happy	**0.126**	*0.353*	0.507
	Sad	*0.237*	**0.098**	0.328
	Angry	*0.317*	0.405	**0.193**

have difficulty modifying into angry.

2.6.2 Survey

In addition to the above, we analyzed the subjective metrics collected in the survey in light of the hypotheses laid out above.

H1

For the first hypothesis, there should be no difference in the recognition accuracy for the target emotions between the original and modified movements. For example, movements modified to be happy should be recognized as happy with the same accuracy as original happy movements. TOST (two one-sided tests) equivalence tests were performed between the original and modified movements for each target emotion. Given the range of the accuracies (0 for wrong, 1 for correct), the equivalence test α was set to 0.1. The results (Table 2.4) show that **H1** is supported ($p < 0.05$) for happy→sad and sad→angry and implied ($p < 0.1$) for angry→sad, but is not supported for the other modifications.

Table 2.4: Mean emotion recognition accuracies and equivalence test p-values (italicized). Bolded p-values support H1.

		Target emotion		
		Happy	Sad	Angry
Original emotion	Happy	0.59, ——	0.63, ***0.03***	0.18, *0.13*
	Sad	0.44, *0.91*	0.66, ——	0.21, ***0.01***
	Angry	0.44, *0.91*	0.61, *0.08*	0.24, ——

H2

For the second hypothesis, there should be no difference in the legibility scores for the target emotions between the original and modified movements. For example, the legibility scores for movements modified to be happy should not be significantly different than the scores for original happy movements. The legibility score is the Likert score for the target emotion, e.g. the legibility score for a sad movement modified to be happy would be the Likert score for the happy slider in Figure 2.6. Equivalence tests between the original and modified movements for each target emotion were performed. Given the range of the Likert scores (1-5), the equivalence test α was set to 0.2. The results (Table 2.5) show that **H2** is supported ($p < 0.05$) for all modifications.

Table 2.5: Mean emotion legibility scores and equivalence test p-values (italicized). Bolded p-values support H2.

		Target emotion		
		Happy	Sad	Angry
Original emotion	Happy	3.33, ——	3.42, ***0.02***	2.07, ***0.02***
	Sad	2.77, ***0.02***	3.27, ——	2.27, ***0.02***
	Angry	2.81, ***0.02***	3.54, ***0.02***	2.26, ——

2.7 Discussion

The objective results show that the network achieves the learning goals well: the reconstructions look similar to the original movements but with less exaggeration. This is in line with challenges reported for generative networks in other domains [174]. It was particularly interesting that the network and regression model were able to map the movements into the circumplex space with minimal domain knowledge apart from the emotion centroid locations. Qualitatively, valence corresponds to looking upwards or downwards while arousal corresponds to exaggeration.

The subjective results show that **H2** is supported for all modifications, but **H1** is only partially supported. Using the dropdown menu for automatic modification may have been a limiting factor, as evidenced by its mediocre cross-emotion performance (Table 2.3). Manually moving the sliders while monitoring the output for fine-tuning may have yielded better modified samples.

The subjective results also imply that not all emotions are equally conveyable. Anger is consistently recognized below the chance level of 0.33, implying that Blossom may have difficulty conveying anger compared to happy and sad. When creating the movements, anger was the most ambiguous while sad movements were usually a slow lowering of the head. This shows the relationship between a robot's expressive capabilities and its embodiment, which renders certain emotions harder to convey. Movements modified to be happy scored considerably lower in terms of both accuracy and legibility than their original counterparts. This implies that this modification method may not be able to retain some qualities of hand-crafted movements. These observations highlight the difficulty in quantifying emotions, especially with the simple circumplex model.

Plans for future work include a usability study for the interface and testing with other robots to evaluate generalizability. The study would assess the ease-of-use of the interface and address the issues with the automated modification. We would also test the method using robots with more complex modalities such as sounds or face gestures. These modalities may better convey emotions that are difficult to express through movements alone. Additionally, we could choose to imbue affect into non-emotive or task-oriented gestures such as hesitating or signaling. We also want to explore using other starting points in the latent space, such as neutral movements or random samples, to generate new gestures.

2.8 Conclusion

We presented a method for modifying affective movements for an expressive robot using neural networks. Using a dataset of hand-crafted movements, we trained a classifying VAE to learn a latent space to compactly represent the movements and classify them by their intended emotions. We then used linear regression to map the abstract latent space into the comprehensible valence and arousal dimensions on the circumplex emotion model. Applying simple arithmetic in the latent space enables us to modify the valence and arousal of the movements. We evaluated this approach with objective and subjective metrics which showed that the method performs well along learning objectives and to some extent supported the hypotheses that the modified movements are comparable to the originals in terms of recognizability and legibility. Compared to heuristic approaches for creating movements, we used little domain knowledge of kinematics and robotics. This suggests that using neural networks for generating robot behaviors is more generalizable and accessible, enabling faster and easier methods for expanding a robot's behavior library for prolonged interaction.

CHAPTER 3

FACE2GESTURE: TRANSLATING FACIAL EXPRESSIONS INTO ROBOT MOVEMENTS THROUGH SHARED LATENT SPACE NEURAL NETWORKS

3.1 Introduction

Robots use movement to convey internal affective states for more compelling human-robot interactions. However, creating movements often requires working knowledge of robotics and kinematics. Even accessible methods such as kinesthetic teaching are constrained by limited access to robots. Relying primarily on retrieving preprogrammed responses from a static database can diminish users' interest in the robot as time goes by [175]. Generating new behaviors in response to different users' idiosyncratic inputs may mitigate this novelty effect and suggest that the robot has a deeper capacity for affective understanding. Machine learning models, particularly deep neural networks, have achieved state-of-the-art performance in a variety of applications, such as emotion recognition [176]. Neural networks have also shown promise in data generation, such as generative adversarial networks for photorealistic images and conversational chatbots [177, 178]. Therefore, we believe that neural networks are well-suited for affective generation applications.

We propose an approach to generating affective robot movements in response to user inputs, specifically emotive human facial expressions. We use a convolutional variational autoencoder (VAE) to compress robot movements into a latent embedding space, and a convolutional image encoder to compress face images into the same latent space. To align the disparate modalities in the shared latent space, we implement a triplet loss objective to cluster embeddings by emotion classes rather than by modality. We evaluated this approach in an online user survey where participants watched the robot performing the generated movements and selected the corresponding image from a lineup. We found that generated happy and sad movements were well-matched, but angry movements were mostly mismatched to sad images.

Our contribution is an approach for translating facial expression images into affective robot movements using neural networks. This approach has further implications for expanding an agent's behavior library and for other multimodal affective applications, e.g. a listening ear responding to text sentiment and audio inflection, or a video-watching companion reacting to the multimodal context of video.

3.2 Related Work

We based our approach upon prior works in robot movement creation and neural networks.

3.2.1 Robot Movement

Movement enables robots to interact with the world with affordances beyond screen- and audio-based agents [66]. Apart from goal-oriented actions such as locomotion or manipulation, movement can also communicate affective states, either in discrete categories (e.g. happy, sad, angry [149]) or on a continuous spectrum (e.g. valence, arousal [4]). However, designing emotive movements requires depth of knowledge in robotics, movement analysis, and affective expression. To reduce the need for hand-crafting behaviors, researchers have explored generating movements using machine learning models [179, 180]. We are interested in generating affective movements using machine learning techniques, specifically neural networks. We view these generated movements as not replacing the user-crafted movements, but rather complementing them and expanding the robot's available behaviors.

3.2.2 Neural Networks

Robot movement generation

Due to the cost of designing robot movements, which is often time-intensive and limited by proximity to physical robots, machine learning models can use existing movements to expand a robot's available behavior library. Marmpena et al. generated motion for a humanoid robot by chaining poses together from a VAE's learned latent space [181, 182]. Yoon et a. generated gesticulation motions for a humanoid robot using a multimodal dataset of speech, text, and posture [183]. The works in this space have largely focused on humanoid embodiments, perhaps due to the familiarity and availability of humanoid movement data. Additionally, these approaches rely on datasets that are either expert-crafted or sourceable in large quantities, e.g. professionally recorded speeches to yield paired multimodal datasets. We adopt similar neural network methods, but instead rely on user-crafted movements. We believe that sourcing movements from users is a more accessible approach and yields samples that better reflect the potential end-users of such a system.

Affective applications

The ability for neural networks to learn features is useful for applications that may otherwise be intractable with heuristics, such as affective recognition and generation. Many works in this space focus on perceptive tasks, such as supervised sentiment analysis in text and images [184, 185]. Neural networks can also generate emotive samples of images and audio [186, 187]. Our proposed application is less technically complex than these examples, particularly in the relatively low dimensionality of movement compared to high-dimensional images, text, and audio.

Multimodal machine learning

The ability for neural networks to learn features is also useful for multimodal applications [188]. Automatic image captioning is a common application that learns alignments within a paired dataset of images and their corresponding textual descriptions [75]. Reversing the task to generate images given text descriptions is a more complex task, but recent state-of-the-art techniques are capable of incredibly realistic generated samples [189]. Nguyen et al. performed manifold alignment on a paired image and text dataset for robot understanding [78]. These techniques are well-matched to the inherently multimodal affordances of robots.

We use techniques from these previous works to create an affective response system that generates robot movements from facial expressions. We perform intermodal translation by using techniques from multimodal machine learning, specifically encoder-decoder architectures and class-based triplet loss. The resulting system encodes both robot movements and human facial expressions into a shared latent embedding space, and decodes these embeddings to generate movements from either modality.

3.3 Methods

We used an existing robot platform, datasets of movements and face images, and encoder-decoder neural networks.

3.3.1 Robot Platform

We used the Blossom robot, as previously described in Section 2.4.1.

3.3.2 Data

Movements

We used robot movement samples that we crowdsourced from lay users. We asked users to first view video prompts of cartoon characters conveying different emotions (happiness, sadness, anger), then to puppeteer the robot with their phones as if it were conveying the same emotion. Some movements were collected locally in-person, though most were collected remotely by users teleoperating the robot. To account for the subjectivity of the user-crafted samples, we filtered the dataset by deploying a survey to another set of users. Each question contained a video of the robot performing each movement, followed by a question asking users to select the conveyed emotion. We deployed the survey through Amazon Mechanical Turk and received over 250 responses, averaging 25 ratings for each movement. We kept only movements recognized at a threshold of 50%, an arbitrary margin above the chance level of 33%. This filtering downsized the original dataset from over 200 samples to approximately 140. We then balanced the emotion classes by oversampling from the smaller class populations. Because the neural network requires fixed-length inputs, we took random 4.8-second samples from each movement. Though we can expand the data through augmentation, we took care to perform only augmentations that are emotionally neutral, e.g. mirroring a movement from left to right is neutral and valid, but modulating the pitch of the robot's head downwards may convey more sadness and is thus invalid. We designed the following augmentations:

- Shearing the DoFs in time by slightly nudging their trajectories relative to each other.
- Mirroring horizontally by swapping the left and right tower motors (2 and 3) and reversing the base rotation.
- Decoupling the left and right tower motors. Because these motors are often synchronized in the user-crafted movements, they have a tendency to collapse into copies of each other. Desynchronizing these DoFs promotes rolling motion.
- Shifting the average base rotation slightly. Because the robot faces directly forward in many user-crafted movements, this augmentation prevents neglecting the base motor and promotes yawing motion.

Face images

We used the Cohn-Kanade dataset, a collection of facial expression videos from a diverse range of actors [76]. We used the final frame at the apex of each emotion, resulting in approximately 150 samples. We augmented the data with low-magnitude rotation, translation, horizontal mirroring, scale, shear, and brightness transformations.

3.3.3 Network

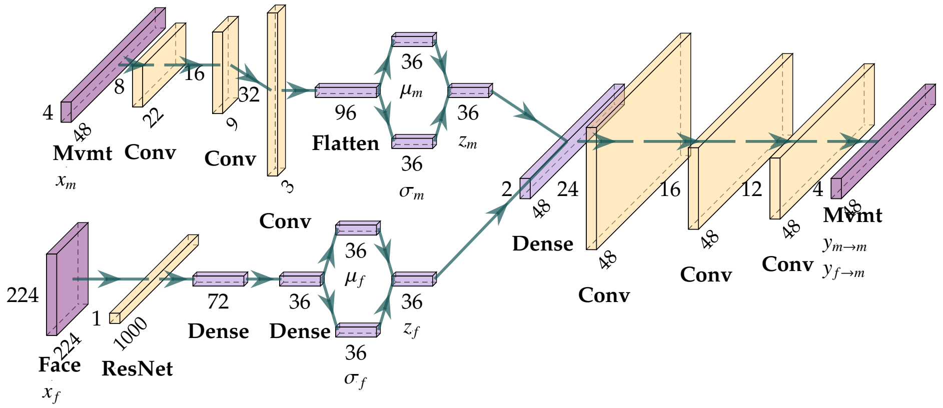


Figure 3.1: Neural network for translating face images into movements. The user-crafted ***m***ovements x_m (4.8 seconds at 10 Hz with four DoFs → 48 × 4) are encoded into a 36D embedding space $z_m \sim N(\mu_m, \sigma_m)$ (top left), then decoded to reconstruct the original input $y_{m \to m}$ (right). The ***f***ace images x_f are encoded into the same 36D embedding space $z_f \sim N(\mu_f, \sigma_f)$ (bottom left), then decoded to generate new movements $y_{f \to m}$ (right).

We constructed the end-to-end network using convolutional encoders and decoders for each data modality.

Algorithm 1: Training algorithm

```
Input : Input movements $X_m$, input face images $X_f$
$F_m(x_m) \leftarrow f_{dec}(f_{enc}(x_m))$ //movement autoencoder neural network;
$F_f(x_f) \leftarrow f_{embd}(ResNet_{50}(x_f))$ //face image encoder neural network;
while not converged do
    $x_m, x_f$ //minibatch of movements and faces;
    $y_{m \to m} \leftarrow F_m(x_m)$ //movement reconstructions;
    $z_m \leftarrow f_{enc}(x_m)$ //movement embeddings;
    $z_f \leftarrow F_f(x_f)$ //face embeddings;
    $L_r \leftarrow MSE(y_{m \to m}, x_m)$ //reconstruction loss with mean-squared error;
    $L_{KL,m} \leftarrow KL(z_m)$ //movement KL divergence;
    $L_{KL,f} \leftarrow KL(z_f)$ //face KL divergence;
    $L_t \leftarrow T(z_m, z_f)$ //triplet loss (Equation 3.1);
    $L \leftarrow w_r L_r + w_{KL,m} L_{KL,m} + w_{KL,f} L_{KL,f} + w_t L_t$ //overall loss, backpropagate to update networks $F_m$ and $F_f$;
    $y_{f \to m} \leftarrow f_{dec}(z_f)$ //pass face embeddings through decoder to generate movements;
end
$F_{f \to m}(x_f) \leftarrow f_{dec}(F_f(x_f))$ //face-to-movement translation network;
```

Movement VAE

We used a VAE to compress the movement data into embeddings in a lower-dimension latent space (Figure 4.6, top left to right). The encoder f_{enc} uses 1D convolutions that stride across the time dimension of the ***m***ovements $x_m \in X_m$, and outputs the latent space distribution parameters (log-mean and log-variance of a distribution $N(\mu_m, \sigma_m)$). We empirically set the latent dimension to 36. The decoder f_{dec} uses these parameters to sample embeddings $z_m \sim N(\mu_m, \sigma_m)$ which pass through deconvolutional layers to reconstruct the original movements $y_{m \to m}$. We used LeakyReLU ($\alpha = 0.1$) and batch normalization after each convolutional and fully connected layer. We calculated the reconstruction loss

L_r as the mean-squared error between the raw trajectories of the original and reconstructed movements. The VAE also uses Kullback-Leibler (KL) divergence as a loss $L_{KL,m}$ to ensure that the embedding distribution approximates a normal distribution, i.e. $N(\mu_m, \sigma_m) \approx N(0, 1)$. Because the embedding space is sampled from this distribution, the trained network can generate new movements by sampling embeddings from $N(\mu_m, \sigma_m)$ and passing them through f_{dec}.

Face image encoder

We encoded the images of ***f***aces $x_f \in X_f$ into the same latent space by first passing them through a pretrained ResNet$_{50}$ model [77], then through two fully connected layers (Figure 4.6, bottom left). Similar to the VAE, we used LeakyReLU and batch normalization after the fully connected layers, and the final encoder layers yield the embedding distribution $z_f \sim N(\mu_f, \sigma_f)$. We added the KL divergence of the face embeddings $L_{KL,f}$ to the overall loss.

Shared latent space alignment using triplet loss

To align the embeddings Z_m and Z_f in the shared latent space, we used triplet loss L_t [78]. The triplet loss minimizes the distance between an ***a***nchor embedding z_a and a positive sample embedding z_+, and maximizes the distance between the anchor and a negative sample embedding z_-. For each sample in a minibatch, we mined positive samples by randomly sampling embeddings that share the same emotion class, and negative samples from the other classes. We used an imbalanced mining scheme wherein movement embedding anchors can sample from either modality, while face embedding anchors only select positive samples from the movement embeddings. The intuition is that the image encoder can easily separate the emotions due to the pretrained ResNet$_{50}$ model and should primarily be fine tuned to match the movement embedding space. For example, given a happy movement as an anchor, positive samples come from happy movements and images, and negative samples come from the set of sad and angry movements and images. However, given a happy face image as an anchor, positive samples come only from happy movements. We used the Euclidean distance function $d(a, b)^2$ with no margin.

$$L_t = \sum_{z_a \in Z_m \cup Z_f} max(d(z_a, z_+)^2 - d(z_a, z_-)^2, 0) \tag{3.1}$$

The overall loss objective of the network is a weighted combination of the reconstruction, KL, and triplet losses:

$$L = w_r L_r + w_{KL,m} L_{KL,m} + w_{KL,f} L_{KL,f} + w_t L_t \tag{3.2}$$

We empirically set the weights as $w_r = 1 \times 10^4$, $w_{KL,m} = 1 \times 10^{-2}$, $w_{KL,f} = 1 \times 10^{-1}$, and $w_t = 1 \times 10^3$.

Algorithm 1 describes the training loop. Due to the subjectivity of the outputs, we both monitored the loss curves and appraised the quality of the generated movements during training. After training, we can use the function $F_{f\to m}(x_f) = f_{dec}(F_f(x_f))$ to translate face images into movements $y_{f\to m}$ (Figure 4.6, bottom left to right). We trained for 1,500 epochs with a learning rate of 1×10^{-2}, batch size of 32, Adam optimizer, and an 80-20 train-test split.

3.4 Evaluation

We evaluated the approach through both objective technical metrics and a subjective user survey.

3.4.1 Network Evaluation

We evaluated the technical performance of the method through its performance in minimizing the loss objectives. We also monitored the outputs: the reconstructed and image-generated movements, and the separability of the latent embedding space. As an ablation study, we analyzed the performance of the network optimizing either only reconstruction loss or only triplet loss.[1]

3.4.2 User Evaluation

Due to the subjective nature of the proposed method's outputs, we performed a user evaluation through an online survey. We constructed a survey where each question showed a video of a movement and a lineup of three images, consisting of the movement's actual source image and two random images sampled from the other emotion classes. We asked users to view the video and select the image that best corresponds to the movement. We defined a baseline as using a source image's known emotion label and randomly selecting a user-crafted movement sample of the same corresponding emotion class, e.g. pair a randomly chosen happy face image with a randomly chosen happy movement

[1] Because KL divergence only helps shape the learned latent space but does not by itself generate movements or align embeddings, we do not ablate for a KL-only configuration.

sample. Rather than claim that our method improves upon the baseline, our method avoids the repetitiveness of recycling a predefined library of behaviors, the benefits of which would require a longitudinal evaluation. Our hypothesis is that the generated movements will be recognized above the 50% level used to filter the dataset (Section 3.3.2). We deployed the survey on Amazon Mechanical Turk and received responses from 50 participants, each of whom viewed the complete set of 30 user-crafted and generated movement samples.

3.5 Results

We analyzed the results through objective technical metrics and the subjective user evaluation.

3.5.1 Network Training

We monitored the reconstruction and triplet losses during training (Figure 3.2). There is a gap between the triplet training and testing loss, indicating overfitting. As explained later, this gap may be a limitation of the network's ability to separate happy and angry movements, particularly those it may not have trained on.

Reconstruction

We evaluated reconstruction quality by comparing the inputs x_m to the outputs $y_{m \to m}$ (Figure 3.3). The outputs capture the overall trajectories of the inputs, but

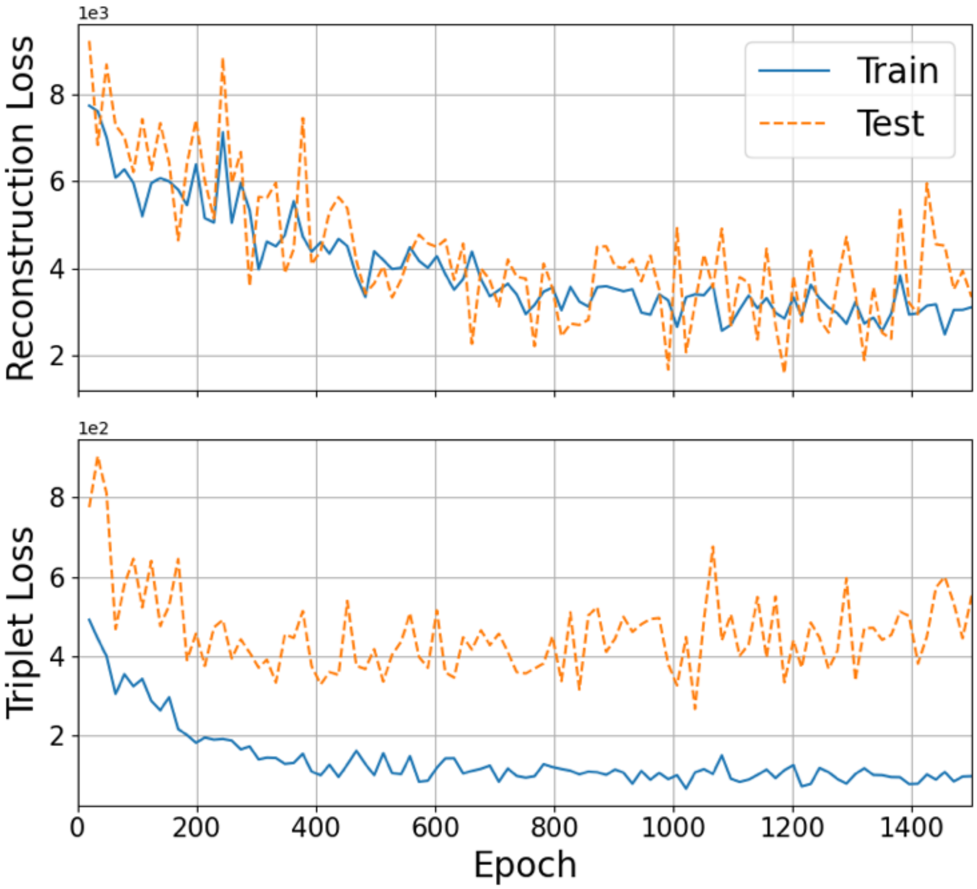


Figure 3.2: Network training curves for reconstruction (top) and triplet loss (bottom). Triplet loss shows signs of overfitting, perhaps due to a coupling of perceptually similar happy and angry movements.

have difficulty preserving exaggeration and tend to smooth out low-amplitude high-frequency "jittering."

Embedding separation

We evaluated embedding separability by visualizing the latent space $Z_m \cup Z_f$ using t-SNE (Figure 4.13, left) [173]. Happy and sad samples are well-aligned, but angry movements are barely separated from happy movements. This coupling may be due to the ambiguity in the data itself (i.e. happy and angry are both

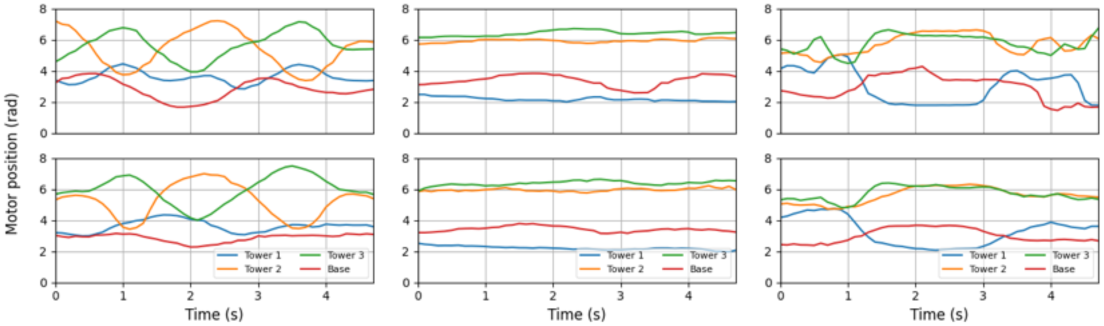


Figure 3.3: Examples of original movements x_m (top) with their reconstructions $y_{m\rightarrow m}$ (bottom) (happy left, sad middle, angry right). Tower 1 controls the pitch of the front of the head, towers 2 and 3 control the left-right rolling of the head, and base controls left-right yaw. The reconstructions maintain the overall trajectories but have difficulty preserving the exaggeration and low-frequency high-amplitude components of the originals.

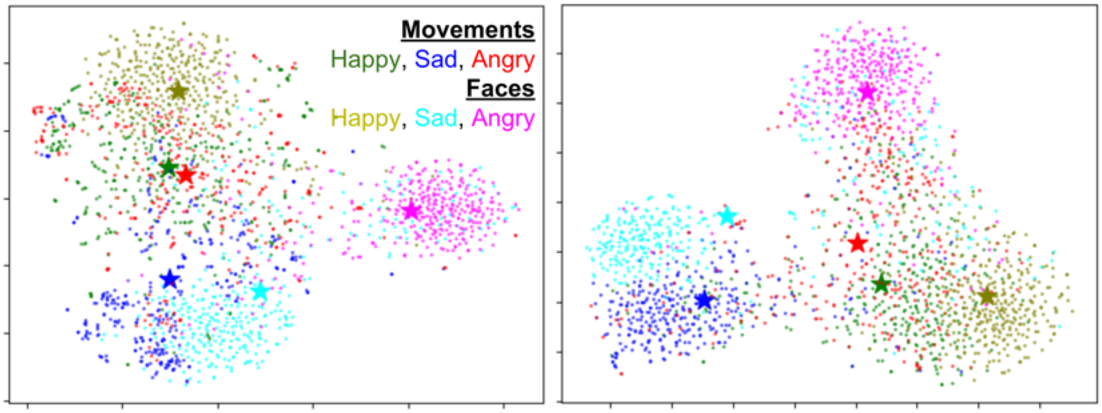


Figure 3.4: t-SNE plots of the shared latent embedding space for the full multi-objective network (left) and a network optimizing only triplet loss (right). Colors indicate modality (movements, faces) and emotion (happy, sad, angry). Stars indicate centroids of each class. Happy and sad movements and faces are closely aligned, but angry movements are barely separated from happy movements, even when optimizing only for triplet loss (right).

high arousal affective states, and are thus difficult to delineate with a simple embodiment), and may also explain the overfitting in the triplet loss training curve (Figure 3.2, bottom).

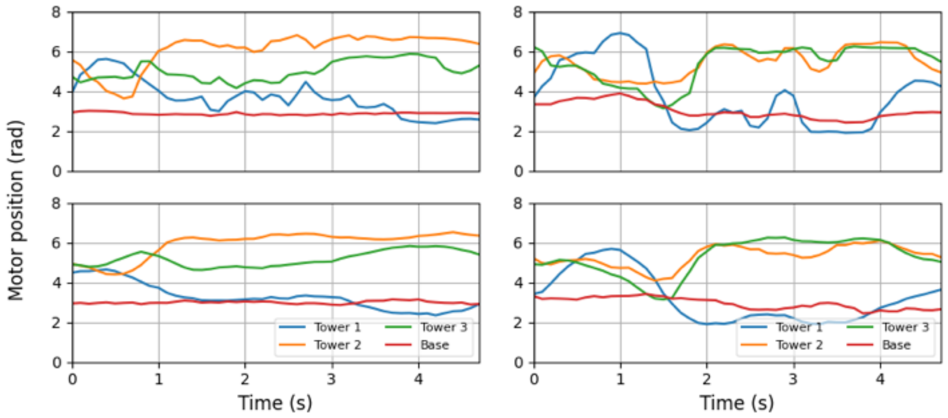


Figure 3.5: Reconstructions for a network optimizing only reconstruction loss. There is only marginal improvement over the standard network (Figure 3.3); exaggeration is slightly better preserved, but jittering is still smoothed out.

Ablation

Using only reconstruction loss defines an upper bound for generating realistic movements, but does not yield noticeable improvements (Figure 3.5). Addressing the deficiencies of the reconstructions (oversmoothed, limited exaggeration) would require alternate techniques such as frequency-domain representation [190, 191].

Using only triplet loss defines an upper bound for the latent space separability (Figure 4.13, right). Even without other objectives, angry and happy movements are still close, suggesting that the coupling is not due to the other losses, but is rather a limitation of the model itself.

Generation

Throughout training, we appraised the subjective quality of image-generated movements $y_{f \to m}$ (Figures 3.6, 3.7). The generated movements retain many of the

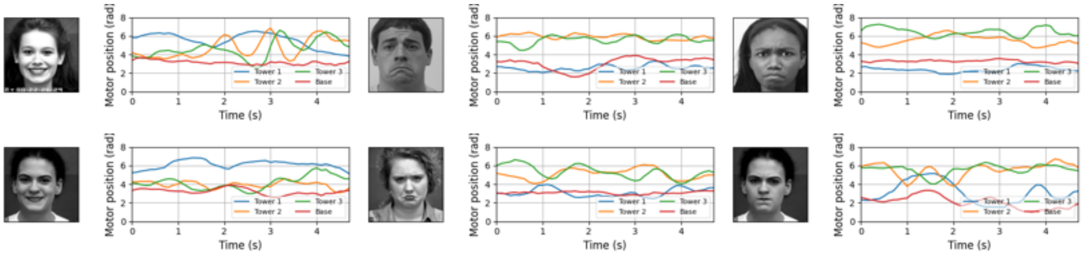


Figure 3.6: Examples of source face images x_f paired with their generated movements $y_{f \to m}$ (happy left, sad middle, angry right). The generated movements maintain similar characteristics of the original user-crafted movements x_m (Figure 3.3), e.g. happy movements have high tower 1 position and sinusoidal out-of-phase rolling motion in tower motors 2 and 3, sad movements have lower tower 1 position and overall flatter motion.

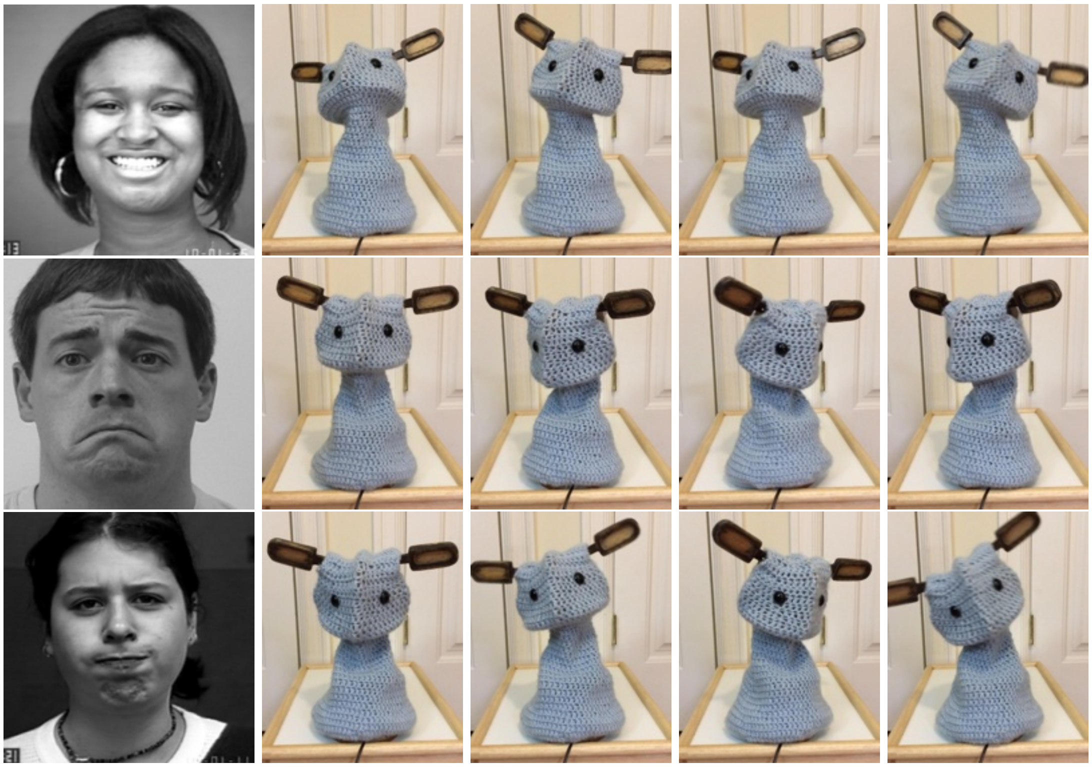


Figure 3.7: Examples of image-generated happy (top), sad (middle), and angry (bottom) movements shown in the survey.

characteristics of the user-crafted movements, e.g. happy movements have high tower 1 position and sinusoidal out-of-phase rolling motion in tower motors 2 and 3, sad movements have lower tower 1 position and overall flatter motion. As with the reconstructions, the generated movements have less exaggeration and jittering than the originals.

Kinematic comparison

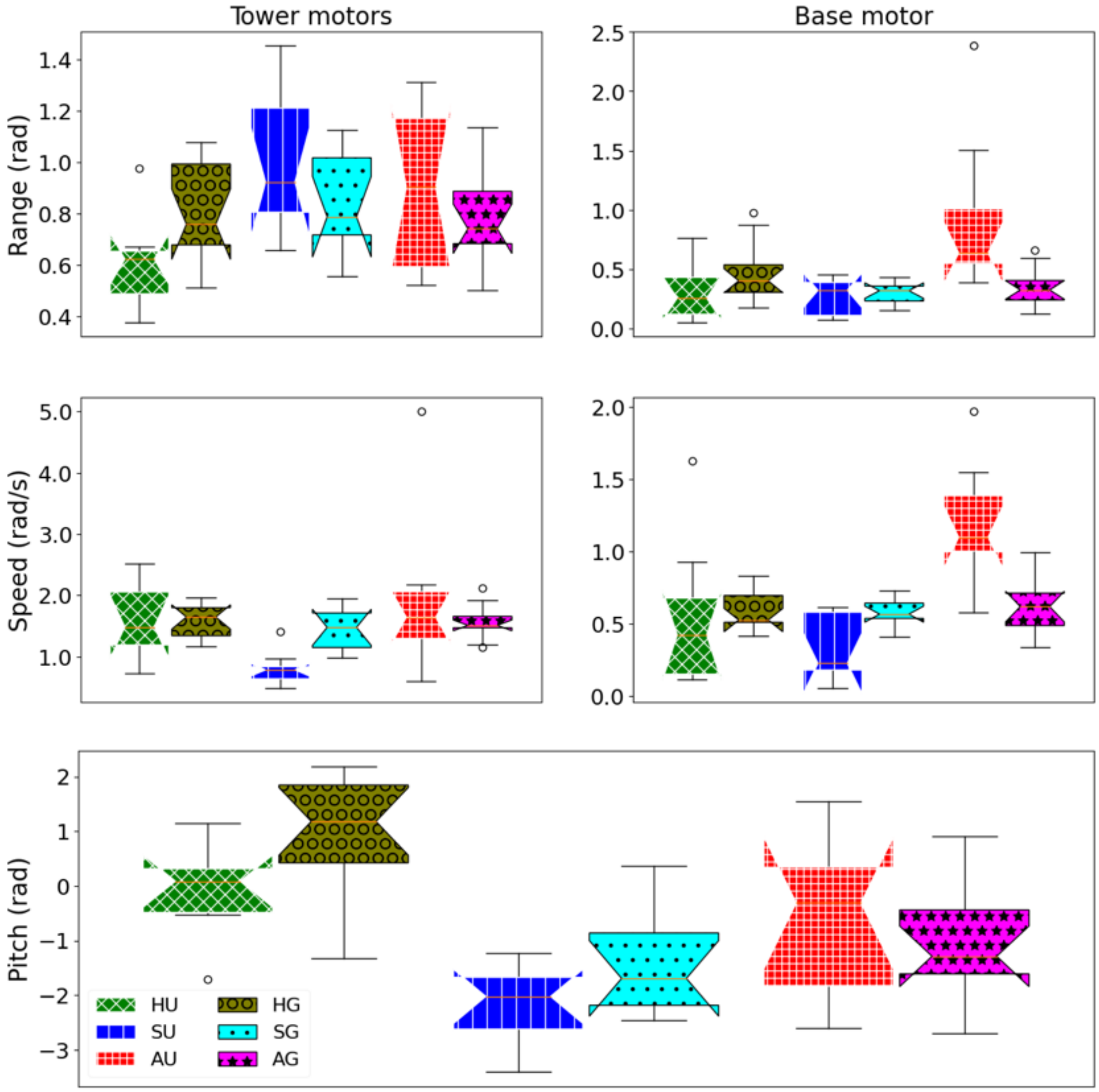


Figure 3.8: Comparison of kinematic features between the user-crafted and image-generated movements. The legend (bottom left) is the emotion (**H**appy, **S**ad, **A**ngry) and source (**U**ser-Crafted, Image-**G**enerated). The user-crafted movements show more between-class variation, but the generated movements preserve many of the overall features.

We compared the user-crafted and image-generated movements from their

Table 3.1: Analytical comparison of the kinematic features (Figure 3.8). The image-generated movements approximate the trends of the means μ of the user-crafted movements, but often have smaller standard deviations σ.

Feature	Source	Happy		Sad		Angry	
		μ_H	σ_H	μ_S	σ_S	μ_A	σ_A
Tower range	User	0.61	0.18	0.99	0.26	0.89	0.30
	Gen	0.79	0.18	0.85	0.18	0.78	0.17
Base range	User	0.31	0.23	0.26	0.14	0.92	0.65
	Gen	0.47	0.23	0.30	0.09	0.35	0.16
Tower speed	User	1.60	0.61	0.80	0.25	1.94	1.25
	Gen	1.61	0.25	1.42	0.33	1.56	0.27
Base speed	User	0.54	0.51	0.32	0.21	1.19	0.40
	Gen	0.58	0.13	0.58	0.09	0.62	0.19
Posture	User	-0.11	0.83	-2.19	0.64	-0.53	1.47
	Gen	0.92	1.04	-1.38	0.90	-1.09	0.97

respective test sets by calculating kinematic features (Table 3.1, Figure 3.8). We calculated range and speed as the peak-to-peak distance and gradient for each DoF, respectively. We calculated pitch as the difference between the positions of the front of the head (tower motor 1) and the average of the sides of the head (tower motors 2 and 3). Positive pitch is looking upwards, and negative pitch is looking downwards. We averaged speed and pitch across the length of each movement. The image-generated movements are mostly comparable to the user-crafted movements, though the user-crafted movements have larger between-class variation (Table 3.1, μ columns), such as the range and speed of the tower motors (Figure 3.8, left column). User-crafted angry movements in particular exhibit noticeably higher base range and speed than their image-generated counterparts (Figure 3.8, right column).

3.5.2 User Evaluation

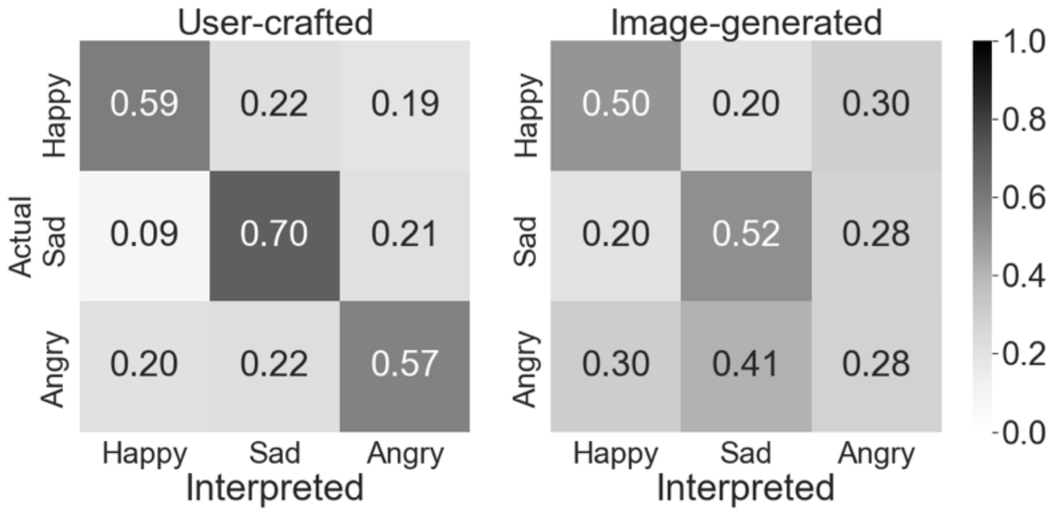


Figure 3.9: Confusion matrices for both the user-crafted (left) and image-generated movements (right). Participants viewed videos of the movement then selected the best corresponding face image from a lineup. While the recognition accuracies for the image-generated movements are lower, happy and sad are still recognized above the 50% level. However, generated angry movements are recognized below chance and are most often mismatched to sad images.

The user evaluation serves as a subjective appraisal of the generated movements. For the survey, we used only data from the respective movement and image test sets, i.e. samples that the network did not train on. For the user-crafted movements, we randomly paired face images only with movements from the movement test set. For the image-generated movements, we used only movements generated from images from the image test set. We used five movements for each condition, resulting in a total set of 30 movements (2 sources × 3 emotions × 5 samples). Each of the 50 survey respondents watched every video, and tried to match the movement to its corresponding source image. We analyzed the user evaluation results with a confusion matrix (Figure 4.16); perfect results would be an identity matrix. The randomly sampled user-crafted movements are overall well-matched (left). The image-generated happy and sad movements are less well-matched (right), but are still above the 50% level we used for filtering the dataset (Section 3.3.2). However, generated angry movements are recognized below chance, being confused primarily for sadness, but also for happiness. To compare the recognition accuracies between the user-crafted and image-generated movements, we performed equivalence tests (two one-sided t-tests) with an equivalence bound of ±10%. These tests yielded *p*-values of 0.39, 0.96, and 0.99 for happy, sad, and angry, respectively, showing that none of the classes are significantly equivalent.

3.6 Discussion

The network training results show that the network is capable of reconstructing the original user-crafted movements and generating new movements from the shared latent space. The difficulty in separating angry movements can be attributed to the limitations of both the model and the platform. Users who created movements noted that it was difficult to convey anger in particular due to the robot's lack of appendages. This limitation may have resulted in angry and happy movements being perceptually similar, as they are both classified as high-arousal emotions on the circumplex model [4]. Additionally, due to the human uninterpretability of the learned embedding feature space and stochastic nature of t-SNE, the 2D visualization may have found more variance in latent features related to arousal and not valence, which could have delineated happy and angry samples.

The confusion of generated angry movements as sad may be attributed to the difficulty in maintaining the exaggeration of the user-crafted movements, as corroborated by the kinematic analysis (Figure 3.8, right column). Though the generated happy and sad movements were recognized above chance, the accuracies were not significantly equivalent to the user-crafted movements. We view the generated movements as not supplanting, but rather complementing existing user-crafted behavior libraries. For example, an agent could use the more legible user-crafted behaviors for "active" scenarios such as call-and-response, while using the generated behaviors for "passive" scenarios such as greeting.

3.6.1 Limitations and Future Work

We used only a subset of the six canonical emotions [149], which themselves are a discretization of the broad continuous spectrum of emotions [4]. This simplification was done in part to reduce the task to the most legible emotions, but also due to the limitations of the limbless robot. Additionally, there may be ambiguity within the image dataset itself. Angry and sad images are both low-valence emotions that may be confounding depending upon the individual performing the expression. This discrepancy is orthogonal to the confusion between angry and happy movements, and highlights disparities between movement and images as affective modalities. Future work could involve using a more expressive platform with more DoFs, expanding the range of emotions and data modalities (e.g. text, audio), and deploying the system in a real-time interactive scenario.

While we achieved good survey results using a between-class lineup, i.e. one image for each of the three emotion classes, the unpaired nature of the different dataset modalities would make it difficult to discern the source image from a within-class lineup, e.g. it would be difficult to confidently select the source happy image from a lineup consisting of only happy images. Although the usability of this approach on unpaired and separately collected data can be seen as a feature, future work would benefit from collecting a paired dataset of prompts and multimodal behavior demonstrations in an attempt to achieve a deterministic translation function.

3.7 Conclusion

In this work, we demonstrated an approach for generating robot behaviors from emotive images using neural networks. We used convolutional encoders to compress affective robot movements and facial expression images into a shared latent embedding space. We used a triplet loss objective to align the multimodal embeddings by emotion, e.g. bringing happy movements closer to other happy movements and faces, and separating them from sad and angry movements and faces. We then used a convolutional decoder to generate movements from embeddings from either modality. Through a subjective user evaluation, we found that happy and sad image-generated movements were recognizable and well-matched to their source images above a 50% level, but generated angry movements were mostly mismatched to sad images. Though the recognition accuracies were not significantly equivalent to the user-crafted movements, the generated movements are still usable for expanding the agent's behavior library. Future behavior systems for affective agents can adopt this approach with different modalities.

Part IV

Telepresence

CHAPTER 4

WHAT IS IT LIKE TO BE A BOT? VARIABLE PERSPECTIVE EMBODIED TELEPRESENCE FOR CROWDSOURCING ROBOT MOVEMENTS

4.1 Introduction

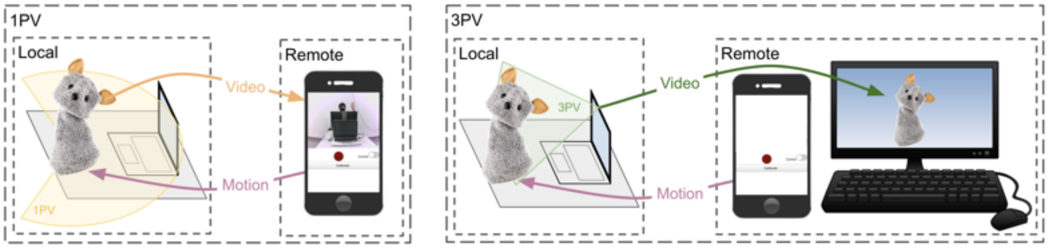


Figure 4.1: In the first-person view (1PV, left) the camera feed is transmitted from the local robot to the remote phone. In the third-person view (3PV, right) the local computer camera feed capturing the robot is transmitted to the desktop. In both cases, the remote phone's motion data is transmitted to the local computer to control the robot's motors.

Social robots can communicate through their embodiment and movements, which serve to not only achieve utilitarian functions but also to convey affective states [66]. Movement is an important nonverbal communication modality that differentiates robots from graphics- or voice-based agents. However, designing robot movements is often a costly process that requires expertise in robotics and movement theory. Accessible methods such as learning from demonstration (LfD) enable lay-users to provide movement samples by either physically manipulating the robot or controlling its degrees-of-freedom (DoFs) [70, 125]. In some cases, larger sample libraries can be elicited using crowdsourcing methods [192, 193, 85]. Movement libraries, whether hand-generated, crowdsourced, or learned, can be further expanded with generative models that analyze existing samples and synthesize new realistic movements [145, 151, 152, 153] (Figure 4.2). For example, deep neural networks can learn important data features

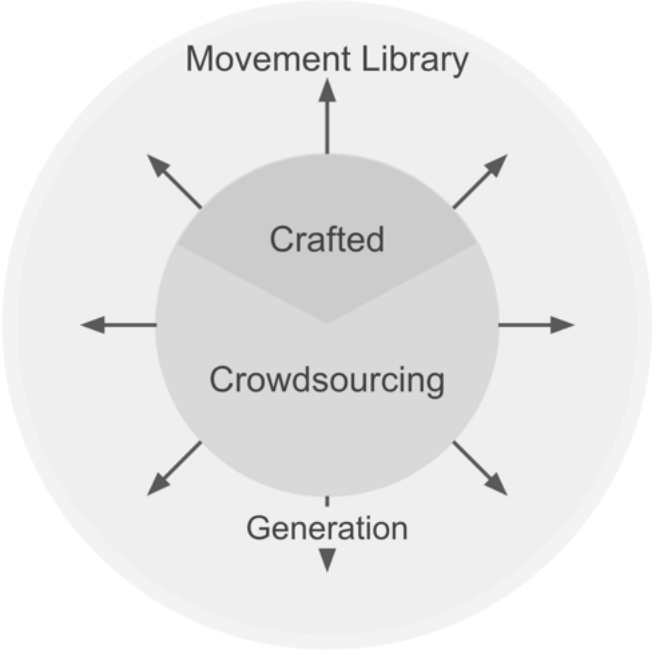


Figure 4.2: Roboticists can complement their initially small set of hand-crafted movements by crowdsourcing new samples from users. Machine learning techniques can then further expand the available movements by generating new samples. This work focuses on the crowdsourcing and generation aspects.

given a sufficient diversity of samples, thus relaxing the need for expert knowledge in movement generation [148]. As a result, human-robot interaction (HRI) researchers have begun applying neural networks for generating robot movements [182, 194, 183], but these approaches are limited by the availability of data.

Restrictions on in-person experiments due to the COVID-19 pandemic forced HRI researchers to shift towards remote technologies, such as simulators or telepresence robots, and this shift could prove beneficial for robot movement generation. Researchers have also used these remote technologies to conduct online evaluations and crowdsource data. Services such as Amazon Mechanical Turk and Prolific have enabled the collection of data from a diverse user base. Paired with telepresence platforms, crowdsourcing could also enable the collection of user-crafted demonstrations for robots. A machine learning model could then use the collected data to generate new samples and further expand the robot's behavior library.

In this work, we present a system for remotely crowdsourcing emotive robot movements through a telepresence robot. The robot is controlled with a smartphone, a widely accessible device that enables a direct mapping from the user's body to the robot using the phone's built-in motion sensors. We compared two alternate viewpoints for the interface: a through-the-robot first-person view (1PV) seen on the phone, and a whole-body third-person view (3PV) seen on an external monitor (Figure 4.1). We performed an evaluation where users controlled the robot and recorded emotive movements to collect a diverse user-crafted data set. To validate the usability of the collected data set for ML movement generation, we trained a neural network to generate new movements, and deployed a survey to subjectively compare the user-crafted and generated movements. Our contributions are:

- An accessible system for remotely motion controlling a robot in either the first- or third-person, requiring no specialized hardware.
- An evaluation of the system as an embodied telepresence platform. We conducted a remote study for users to control the robot, create emotive movements, and rate their experience using the platform comparing the first- and third-person views.
- An evaluation of the quality of the user-crafted movements as a data set for ML generation, first by using a generative neural network to synthesize new movement samples, then by deploying a survey to compare the user-crafted and generated samples.

4.2 Related Work

4.2.1 Affective Telepresence

The physicality of objects can promote nonverbal and ludic interactions beyond the affordances of visual or auditory communication modalities. Strong and Gaver's Feather, Scent, and Shaker were minimally expressive home objects for technologically mediated sociality between remote users [195]. More specifically within robotics, Goldberg's early telepresence robots emphasized playful interactions, such as tending to a garden or uncovering treasures in a sandbox by remotely controlling a robot through the internet [79]. Sirkin and Ju found that augmenting a screen-based telepresence robot with motion improved the sense of presence on both ends [82]. Tanaka et al. compared video, avatar, and robot communications and found that the presence and movements of a robot improved the conversation partner's sense of social presence [196]. The teddy bear Huggable robot enabled remote users to control its gaze and appendages through a web interface [197]. Gomez et al. used the Haru robot for transmitting ”robomojis,” emojis that are embodied by the robot's motion, animations, and sounds [198]. The MeBot telepresence robot features controllable appendages in addition to a screen displaying the remote user [199]. Similarly, Tsoi et al. created a phone application to turn the Anki Vector robot into a telepresence platform controlled with game-like touchscreen joysticks; this work was a direct response to the sudden isolation of children due to COVID-19 safety restrictions [200]. While these embodied platforms afford an additional dimension of engagement beyond virtual agents, using button- or joystick-centric controllers abstract remote users away from their own bodies as a communicative medium.

Motion Control

Rather than use text inputs or game controllers as proxies for controlling robots, proprioceptive motion controls afford a more direct translation between the embodiments of the user and robot, enhancing the sense of self-location and agency [83]. Ainasoja et al. compared motion- and touch-based smartphone interfaces for controlling a Beam telepresence robot, and found that users preferred a hybrid motion-touch interface (motion for left-right steering, touch for forward-reverse) [81]. Jonggil et al. compared touch and motion controls for a mobile camera robot, and found that motion controls improved the user's sense of presence, synchronicity, and understanding of the remote space [201]. In a more affective application, Sakashita et al. used a virtual reality system with head and arm tracking to remotely embody and puppeteer robots [84]. Many of these robots were utilitarian in design and function, and the user perspectives were constrained to first-person views.

Viewpoint Control

In traditional video chat applications, the remote user's view is controllable only by their interaction partner. Müller et al. created a panoramic stitching application to enable remote users to freely adjust their view by panning their phone around the environment, and found that this significantly improved measures of spatial and social presence and slightly improved copresence [80]. Tang et al. extended this work by replacing the panoramic stitching with a 360° camera [202]. They recommended improvement to collocation, such as indicators to dictate gaze direction or ways to convey remote gestures. Young et al. combined the panoramic stitching and 360° camera into a single evaluation while also adding the user's hand into the shared view as a gesture indicator, and found that both implementations increased spatial presence and copresence [203]. Free choice between first- and third-person is a common interface setting in video games, and several works have shown that first-person perspectives increase immersion and the sense of body ownership while third-person offers heightened spatial awareness [204, 205, 206, 207].

4.2.2 Crowdsourcing Demonstrations for Robots

Robotic systems can implement LfD systems that enable lay-users to provide high-fidelity data for machine learning models. However, collecting demonstrations is still time-consuming and often constrained by physical proximity to a robot. Mandlekar et al. created a system for remotely crowdsourcing grasping task demonstrations for simulated and physical robot arms, and found that more data improves model performance [85]. Among various input devices ranging from mice to virtual reality controllers, they found smartphones to be the best compromise of accessibility and functionality. The primary performance metric was grasp success, with completion time as a secondary measure. Timing is an important feature for affective expression, specifically the arousal dimension on the circumplex model of emotions [4]. Rakita et al. found that while users could adapt to a teleoperated robot's physical slowness, latency between the user's movement and the robot executing the motion reduced performance, further emphasizing the importance of timing [208].

There are several gaps in existing works. Prior works focused primarily on the usability of different control methods, but were either constrained to first-person perspectives or designed for utilitarian, nonaffective functions. Alternatively, we are interested in fixing the control input and instead varying the viewpoints. Although prior works measured subjective experiential responses from the users as both operators and interactors with the robot, many did not focus on the affective quality of the robot's movements. Additionally, there are few prior works in enabling remote crowdsourcing of robot movement demonstrations. We address these gaps by designing a robot telepresence system with accessible motion controls and variable viewpoints. We perform user evaluations to assess the subjective usability of the system for creating emotive robot movements. We then use the movements to train a neural network to generate new movement samples, and perform another evaluation to compare the user-crafted and generated movements. This work probes the following research questions:

- Would affective telepresence be better achieved with a first- or third-person perspective?
- Are crowdsourcing movement demonstrations and generative neural networks viable methods for expanding a robot's behavior library?

4.3 Technical Implementation

In this section we detail the technical implementation of the system, including the robot and user interfaces.

4.3.1 Robot

We used the Blossom robot, as previously described in Section 2.4.1.

4.3.2 User Interfaces

To bolster the system's accessibility, we built the application as a mobile browser experience instead of creating a standalone application. This enabled us to iterate quickly and access a rich library of functionality through APIs while obviating the need for external downloads on the user's device. We created two interfaces to accommodate the two viewpoints (Figure 4.1, right): a mobile interface showing 1PV from the camera in the robot's head, and a desktop interface showing 3PV from the host computer's webcam. Users access both interfaces from a public URL.

Mobile interface

The mobile interface consists of a video feed showing 1PV and a simple layout of buttons for controlling the robot (Figure 4.1, center). The layout was inspired by existing controlling and recording interfaces, such as camera applications and voice recorders. Control of the robot is toggled with a slider switch. Users can record and save movements with a large microphone-style recording button. The robot can be reoriented using a calibration button; this resets the robot's yaw orientation relative to the phone's current compass heading, setting it to face towards the external camera. If the user rotates to the endpoints of the base RoM, indicator arrows appear on the interface to direct the user back towards the center.

Desktop interface

The desktop interface consists of a video screen showing 3PV (Figure 4.1, right). For the evaluation (described later in Section 4.4.1), the interface also features a YouTube video player, controls for displaying a video from a given URL, and a Qualtrics survey at the bottom of the page.

4.3.3 Back End

Communication

The robot is connected to the host computer, which also serves the interfaces. We use `ngrok` to enable communication across the internet from the user to the host computer and robot[1]. We open two `ngrok` tunnels: one for accessing the user interfaces, and another for transmitting the phone orientation data to motion control the robot.

Motion control

Kinematic models of the phone and robot translate the phone's orientation into the angular poses of the robot's head (Figure 4.3). The mobile interface uses the `DeviceOrientation` API to report motion events[2]. The phone's inertial

[1] `https://ngrok.com/`

[2] `https://www.w3.org/TR/orientation-event/`

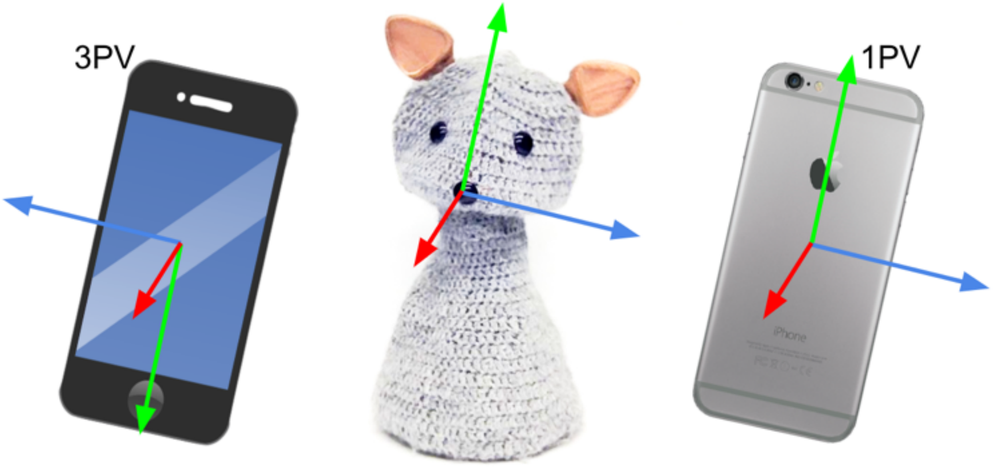


Figure 4.3: The alignment of the robot and phone reference frames when controlling in 1PV. In 3PV, the motion is mirrored to accommodate the perspective of looking straight at the robot (e.g. motion towards the phone's left moves the robot to its right).

measurement unit (IMU) records its pose as Tait-Bryan angles about the phone's reference frame. In 1PV, the phone and robot axes are aligned as if the phone's camera were looking through the robot's eyes. When switching from 1PV to 3PV, the motion is mirrored horizontally to accommodate the front-facing view of the robot, as if the user were facing a physical mirror. In 3PV, yawing or rolling the phone to the left from the user's perspective moves the robot to its right, and vice-versa. Assuming a stable connection, motion data is transferred at a rate of approximately 10 Hz.

Video

For the video streams, we use `WebRTC`, the standard for online audiovisual communication[3]. `WebRTC` manages the handshaking for broadcasting the local video stream to remote viewers.

[3] `https://www.w3.org/TR/webrtc/`

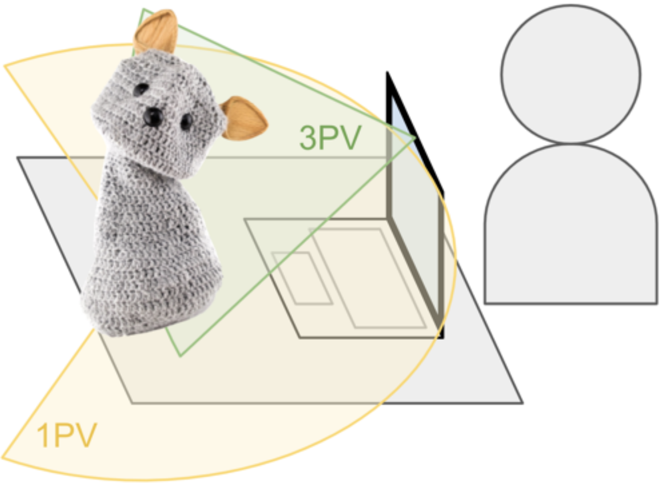


Figure 4.4: Evaluation setup showing the fields of view of 1PV (yellow) and 3PV (green). The evaluation proctor (right) acts as a focal point when controlling the robot in 1PV.

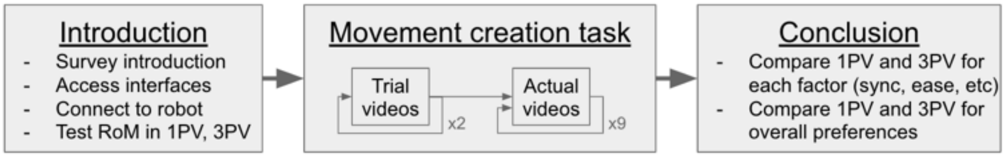


Figure 4.5: Interface evaluation flow. Users first access the interfaces and test the robot's motion. In the main movement creation task, users watch videos of cartoon characters emoting, then create movements for the robot corresponding to the conveyed emotions (happy, sad, or angry). The evaluation concludes with a comparative assessment of the perspectives for user experience factors and overall preferences.

4.4 Experiments

4.4.1 Interface Evaluation

We measured the usability of the system and compared 1PV and 3PV through an online user evaluation for a movement creation task (Figures 4.4, 4.5). We recruited participants through the Prolific online survey platform[4]. We first instructed the user to navigate to the interfaces on both their phone and desktop.

[4] https://www.prolific.co/

Table 4.1: Interface evaluation survey questions, displayed after every video and again at the end of the survey to compare 1PV and 3PV. *Note: scales for mental and physical tiredness are reversed from how they were displayed in the evaluation (1 = not tiring, 7 = tiring) to better match the other factors.*

Question	1 (low rating)	7 (high rating)
How synchronized with the robot did you feel?	Unsynchronized	Synchronized
How much did you feel present in the remote location?	Separate	Present
How easy was controlling the robot?	Difficult	Easy
How enjoyable was controlling the robot?	Not enjoyable	Enjoyable
How engaging was controlling the robot?	Not engaging	Engaging
How mentally tiring was controlling the robot?	Tiring	Not tiring
How physically tiring was controlling the robot?	Tiring	Not tiring
How do you feel about the quality of the movement you created?	Low quality	High quality

The user connected to the robot and tested the controller by looking around the environment in 1PV, then in 3PV. Only one viewpoint (1PV on the phone, 3PV on the desktop) is visible at a time. Because of the importance of timing for the task, we measured the latency between when the orientation data packet is sent from the user's phone and when it is received by the robot's host computer. This latency is only "one-way" and is exacerbated by the video latency, so the user will experience a longer delay from their perspective. Latency below 100 ms is very good and around 1,000 ms (1 second) is serviceable, but exceeding 2,000 ms (2 seconds) noticeably degrades usability. If the user's latency exceeded the 2 second threshold, we would end the study prematurely and compensate the user proportionally to their time spent.

For the main movement creation task, we had the users record examples of emotive gestures. We prompted the users with short videos, between five and ten seconds in length, of a cartoon character (SpongeBob, Pikachu, or Homer Simpson) displaying either happiness, sadness, or anger. We then had the users control the robot to express the emotion from the video and record the movement. We urged users to not simply mimic the motion of the characters, but rather to move the robot as if it were conveying the overall emotion from the scene. Users could rehearse and re-record the movements until they were satisfied, but could not redo the movement once they moved on to the next video. We introduced two trial videos to acclimate the user to the task, followed by nine actual videos (three emotions for each of three different characters). To account for learning effects, we randomized the video orders and perspectives so that each would be equally represented (e.g. four 1PV and five 3PV, or vice-versa). We measured the latency during recording for post-analysis of its effect on the user experience.

We used surveys throughout and after the evaluation to collect user-reported metrics. After each video, we asked for subjective seven-point Likert scale responses to measure experiential factors (Table 4.1). After all of the videos, we again asked for Likert scale responses for each factor, but asked for comparative responses for both 1PV and 3PV. We also asked for overall preferences between the perspectives and included a free response field for any additional feedback. Due to the limited expressiveness of the robot platform, we expected differences across the different emotion classes (e.g. sadness will be more homogeneous but easier to convey than anger). We preregistered hypotheses regarding the experiential factors[5]:

H1.1 1PV will increase the sense of synchronization with the robot due to

[5] Preregistration link: `https://aspredicted.org/pu8p3.pdf`

a heightened sense of embodiment.

H1.2 1PV will increase the sense of presence in the remote location due to higher immersion.

H1.3 3PV will be easier to use due to heightened spatial awareness.

H1.4 1PV will be more enjoyable due to being a unique experience.

H1.5 1PV will be more engaging due to having to move around in one's physical space.

H1.6 1PV will be more mentally tiring due to having to embody a remote system with latency.

H1.7 1PV will be more physically tiring due to having to move one's whole body to maintain a view of the video.

H1.8 3PV will increase the self-reported quality of created movements due to being able to see the full robot.

We enrolled 30 participants through the Prolific platform and offered $10 USD as compensation. We prescreened by participants with access to both a mobile device and desktop. In the interest of minimizing latency, we restricted enrollment to participants living in the United States. We proctored the evaluation through an audio-only Zoom call and took approximately 30 minutes to complete: 10 minutes for the introduction and 20 minutes for creating the movements. We occasionally encountered incompatibilities with certain Android devices, often stemming from access permissions for the orientation sensor. In cases where we were unable to troubleshoot the problem, we ended the study prematurely and compensated the participants proportionally to their time spent; this led us to eventually prescreen to users with Apple devices. We did not have to reject any participants on the basis of high latency.

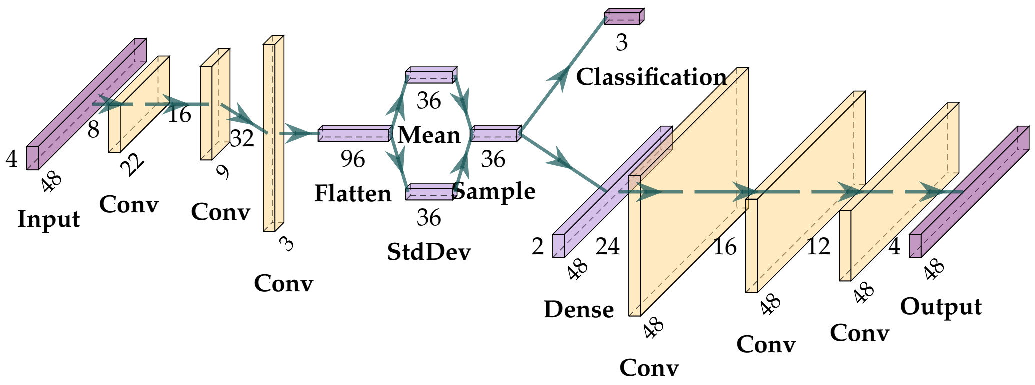


Figure 4.6: Neural network architecture for generating movements. The user-crafted movements (4.8 seconds at 10 Hz with four DoFs → 48 × 4) are used as inputs and encoded into a 36D embedding space (left). The embeddings are both decoded to reconstruct the original input (bottom path) and classified into one of the three emotion classes (happy, sad, or angry) (top path).

4.4.2 Movement Kinematic Evaluation

We calculated kinematic features for each movement: length, speed, and range. Length is the overall duration of the movement, measured in seconds. Because there may have been delays between when the user pressed the record button and actually began or stopped moving, we trimmed the "whitespace" of no motion at the beginning and end of each movement. Speed is the angular velocity of the motors, measured in radians per second. Range is the wideness of the motion in each DoF, measured in radians. We averaged the speed and range across all DoFs for the entire movement. We preregistered hypotheses for the movement features:

H2.1 1PV will yield longer movements due to having to move around in one's physical space.

H2.2 3PV will yield faster movements due to requiring less full-body motions.

H2.3 3PV will yield wider, more exaggerated movements due to requiring less full-body motions.

4.4.3 Dataset Evaluation

To appraise the validity and usability of the system as a data collection platform, we used the user-crafted movements to train a neural network to generate new movements. The network architecture consists of a convolutional variational autoencoder (VAE) with an additional emotion classifier (Figure 4.6) [71]. The VAE encodes the movement samples into a compressed lower-dimension latent embedding space (Figure 4.6, left), then decodes these embeddings back into a reconstruction of the original samples (Figure 4.6, bottom path). The classifier operates on the embeddings and separates the latent space by emotions (happy, sad, or angry) (Figure 4.6, top path). We split the collected dataset by perspective (1PV and 3PV) and trained the network with identical parameters on both subsets. The technical results can be objectively evaluated in terms of the network training metrics, quality of the movement reconstructions, and separability of the emotion classes in the latent embedding space.

We compared the user-crafted and generated movements in a survey to appraise realism, emotiveness, and emotional legibility. We recorded the robot performing the movements from an external perspective similar to 3PV in the first evaluation, and thus used only movements created or generated with the 3PV dataset. We randomly selected subsets of user-crafted movements from a held-out test set and generated movements from the neural network. To avoid using several similar or static movements, we further manually curated the

Table 4.2: Movement comparison survey questions for comparing the user-crafted and generated movements.

1 (low rating)	7 (high rating)
Fake Emotionless	Natural Emotional
Please select the emotion that best describes the robot's movement	Happy, Sad, or Angry

movements to four diverse and representative examples for each condition, resulting in a set of 24 movements (3 emotions × [User, Generated] × 4 examples). Users watched the movements and gave ratings for realism, emotiveness, and which emotion was conveyed (Table 4.2). We preregistered hypotheses for the movement comparison:

H3.1 The generated movements will be as realistic as the user-crafted movements.

H3.2 The generated movements will be as emotive as the user-crafted movements.

H3.3 The generated movements will be recognized with the same accuracy as the user-crafted movements.

4.5 Results

4.5.1 Interface Evaluation Results

We used two-sided t-tests to test **H1** from the end-survey Likert scale responses, and found that many results were significant in the *opposite* direction of our

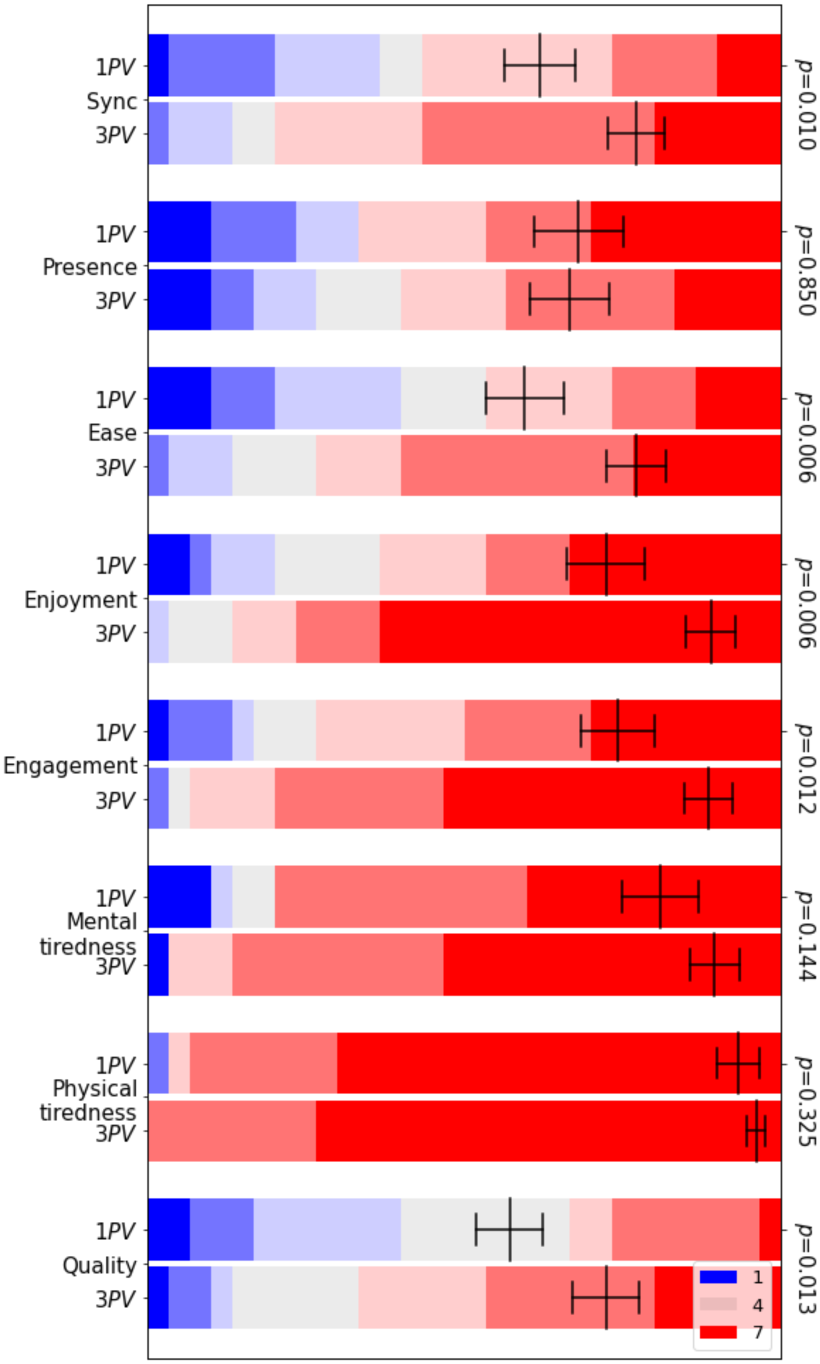


Figure 4.7: Likert scale responses from the interface evaluation end-survey questions. Color indicates level: blue = 1 (low), gray = 4 (neutral), red = 7 (high). Width indicates proportion of responses for a given level. Black bars indicate means and standard deviations. *p*-values of **H1** tested with two-sided t-tests are displayed on the right, and the means indicate preferences for 3PV in all factors except presence and tiredness. *Note: as in Table 4.1, the scales for mental and physical tiredness are reversed from what was displayed in the survey.*

Table 4.3: *p*-values of **H1** tested with two-sided t-tests within each emotion, calculated from the average of the scores after each video. Slight support is suggested only for sadness being more physically tiring in 1PV.

Factor	Happy	Sad	Anger
Sync (1PV>3PV)	0.706	0.995	0.681
Presence (1PV>3PV)	0.365	0.667	0.911
Ease (3PV>1PV)	0.428	0.665	0.430
Enjoyment (1PV>3PV)	0.750	0.637	0.558
Engagement (1PV>3PV)	0.881	0.382	0.630
Mental tired (3PV>1PV)	0.567	0.960	0.619
Physical tired (3PV>1PV)	0.938	0.088	0.718
Quality (3PV>1PV)	0.908	0.744	0.609

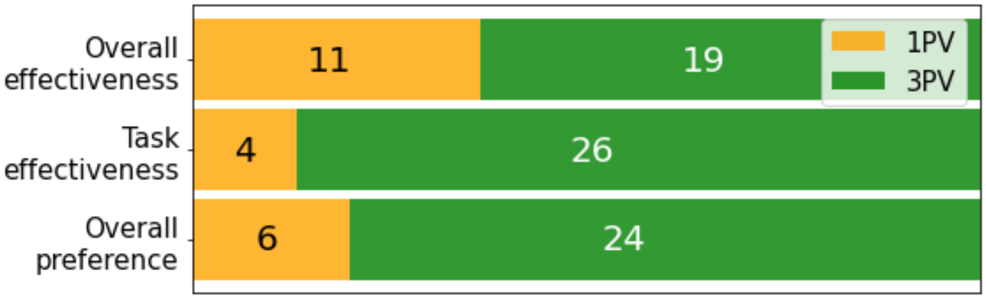


Figure 4.8: Overall preferences reported at the end of the evaluation, showing strong preferences for 3PV.

hypotheses favoring 1PV (Figure 4.7). We found overwhelming preference for 3PV, with significant results in synchronization, ease, enjoyment, engagement, and quality. Even increased presence, which we assumed would be decisively in favor of 1PV, is not supported. We also tested the hypotheses within each emotion class using the responses after every video, and only found slight support for sadness being more physically tiring in 1PV (Table 4.3). Interestingly, the within-emotion scores do not correlate with the comparative end-survey scores. The overall preferences are also favorable toward 3PV (Figure 4.8).

We compared the end-survey scores against the average latencies for each user and for each perspective to analyze latency's effect on the experience (Fig-

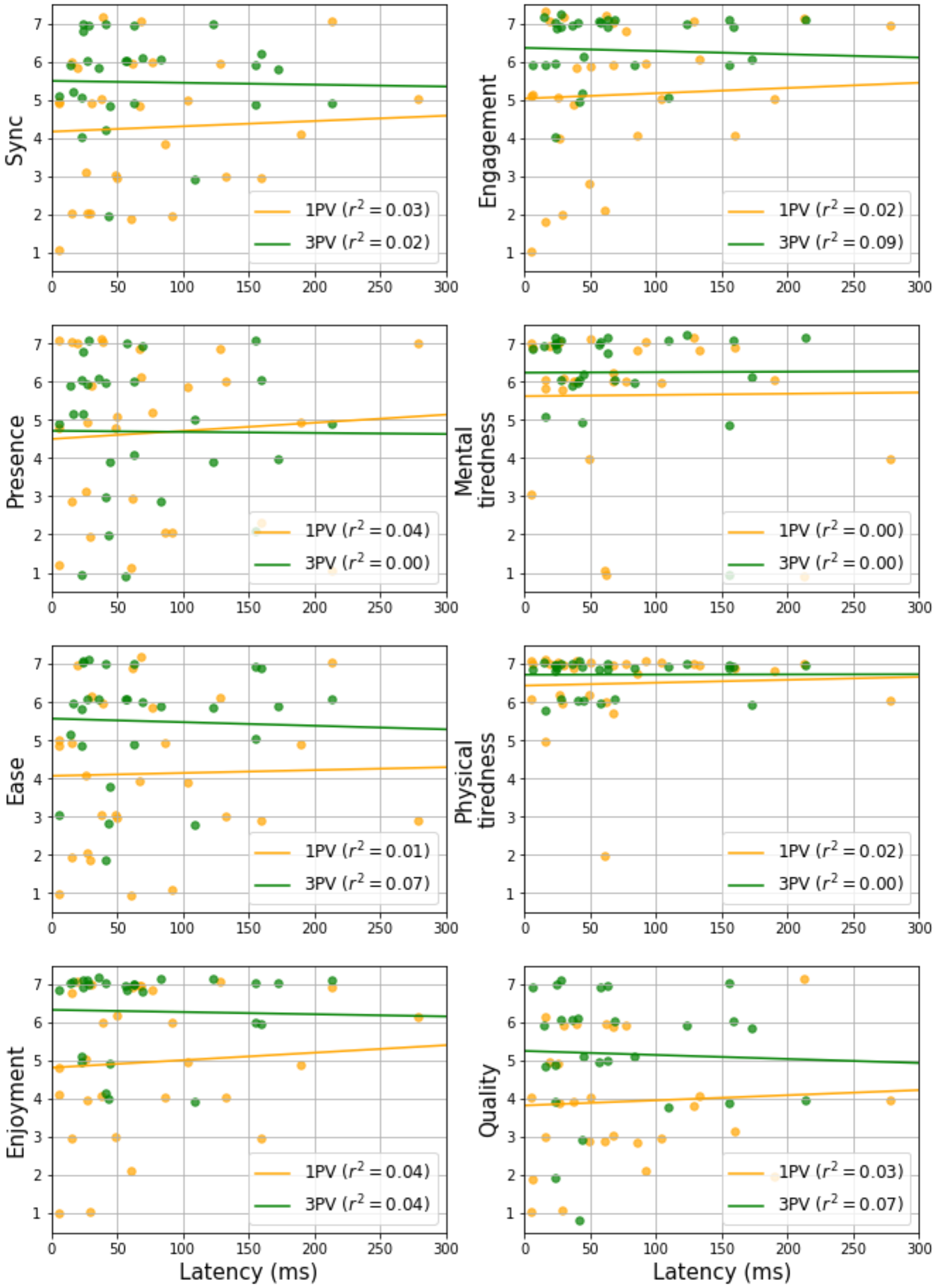


Figure 4.9: Interface evaluation scores versus latency for each user for each perspective. The horizontal axes are truncated to 300 ms (maximum 900 ms) and vertical jitter is applied for legibility. The low r^2 values suggest no correlation between latency and any of the experiential factors.

ure 4.9). As suggested by the low r^2 values, we found no correlation between latency and any factors, suggesting that latency did not noticeably affect the user experience.

4.5.2 Movement Kinematic Evaluation Results

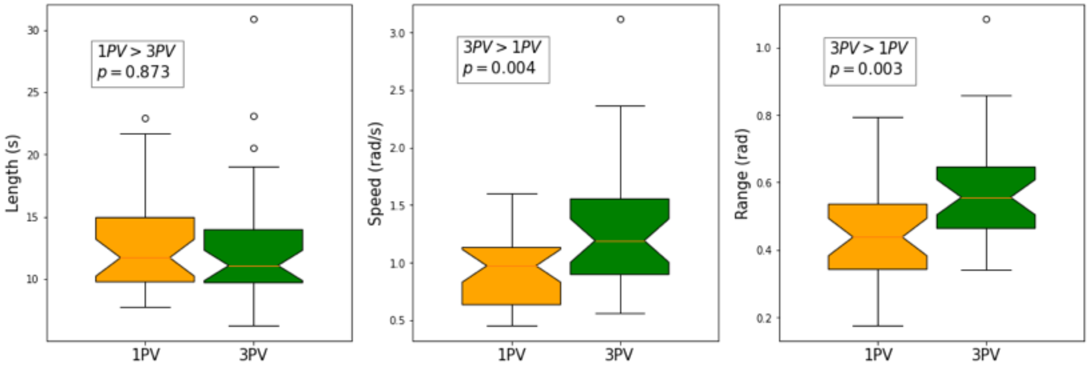


Figure 4.10: Comparison of kinematic features between 1PV and 3PV, testing **H2** with two-sided t-tests. Movement length did not significantly vary between perspectives, but 3PV yielded faster and wider movements compared to 1PV.

We computed the average kinematic features for each user and for each perspective, and used two-sided t-tests to test **H2** (Figure 4.10). We found support for 3PV yielding faster and wider movements, but no support for 1PV yielding longer movements.

4.5.3 Dataset Evaluation Results

The interface evaluation yielded approximately 135 movement samples from each perspective. We prepared the data by chunking the 4-DoF 10 Hz movements into samples of 4.8 seconds with a sliding window of 0.3 seconds, resulting in 48 × 4 data samples. We then performed an 80-20 train-test split and aug-

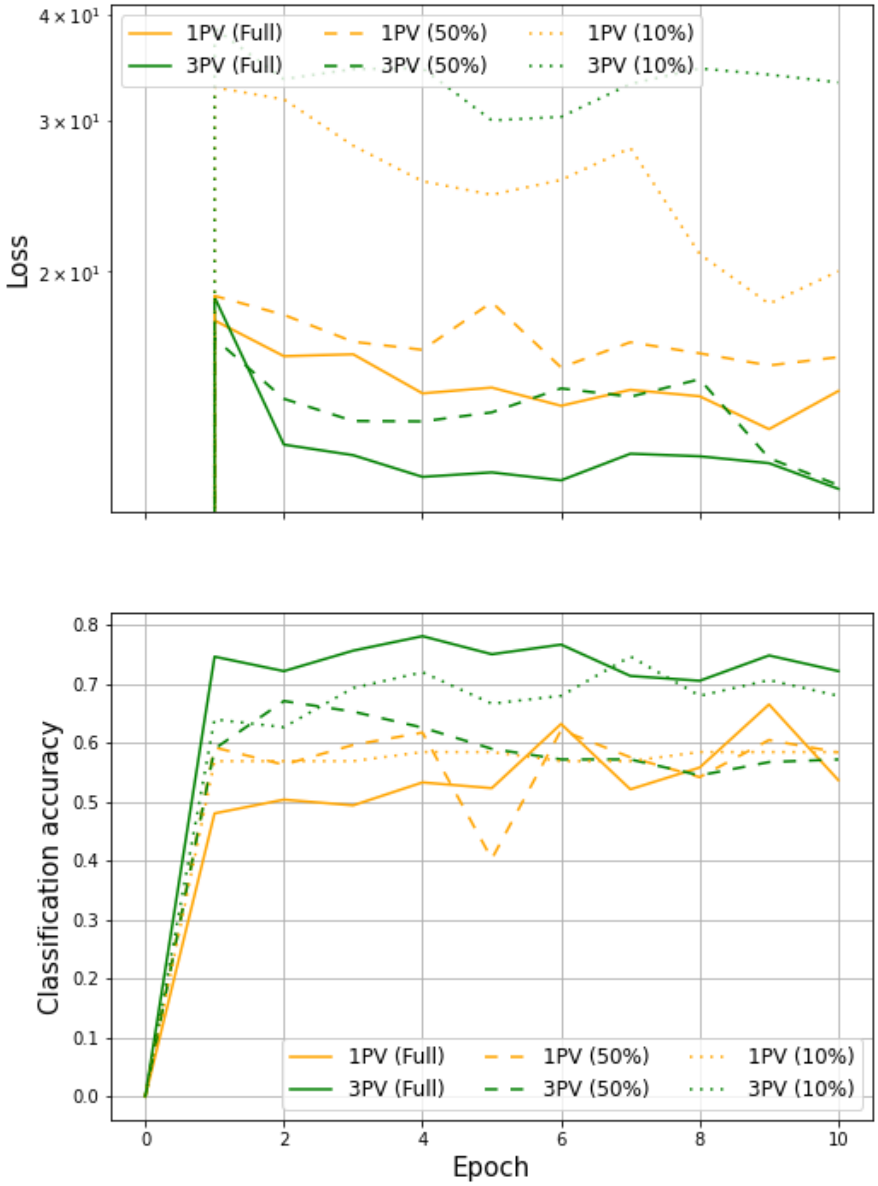


Figure 4.11: Network training results on the test sets. Color indicates perspective, line style indicates data size. Using more data generally lowers the overall loss (top), but only slightly improves classification accuracy (bottom). The small improvement indicates that the network "overfits" to the smaller test set when using less data.

mented the training data by mirroring (flipping left-right), shearing (nudging the timing of DoFs relative to each other), shifting the center (adding small variation to the left-right direction that the robot is looking), and decoupling the left

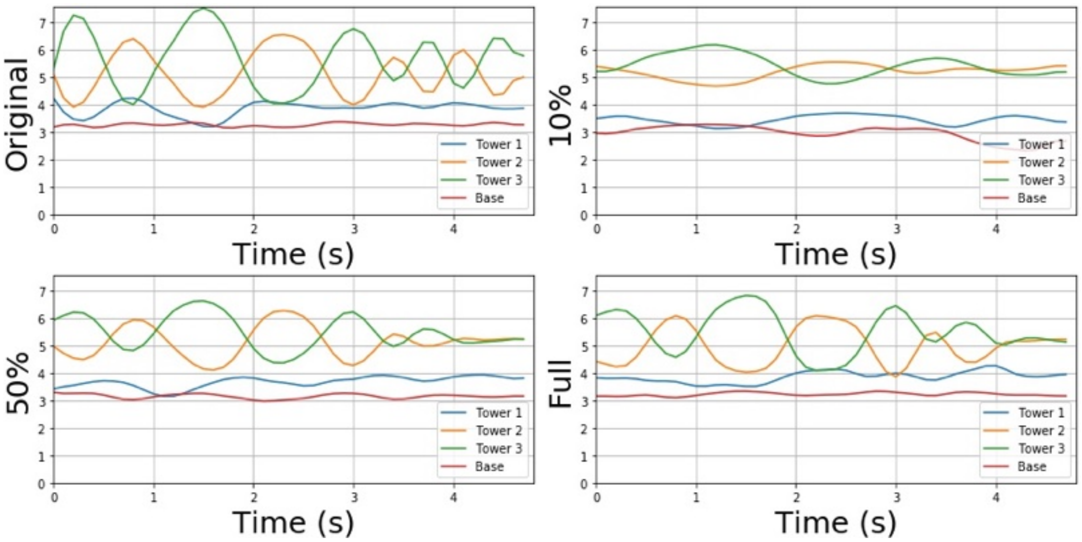


Figure 4.12: Movement reconstructions with varying 3PV dataset sizes. Reconstruction fidelity is proportional to dataset size.

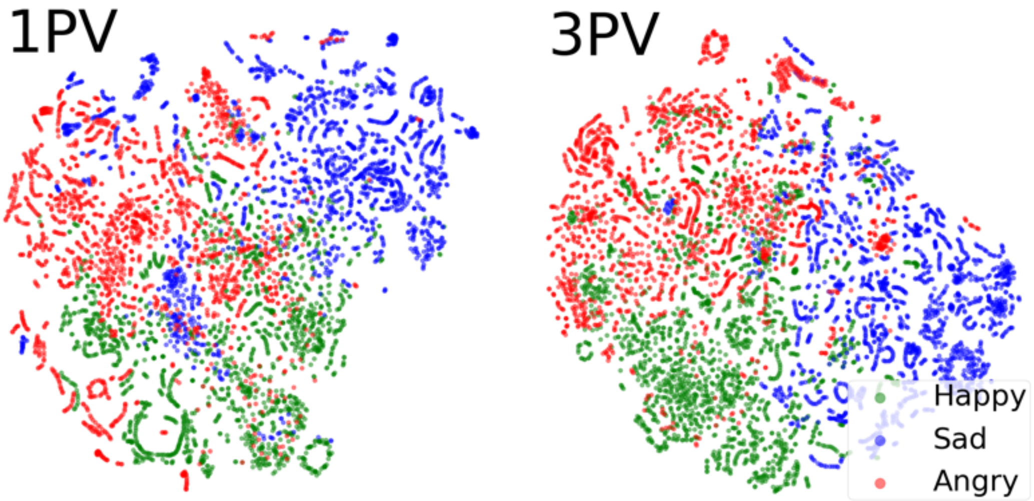


Figure 4.13: Embedding space visualization using t-SNE for 1PV (left) and 3PV (right), color-coded by emotion (happy = green, sad = blue, angry = red). 3PV is more separable, suggesting more diversity and legibility.

and right tower motors (preventing these DoFs from copying each other), yielding over $150,000$ training samples for each perspective. We tuned the neural network architecture and parameters until satisfactory results could be achieved on the datasets from both perspectives.

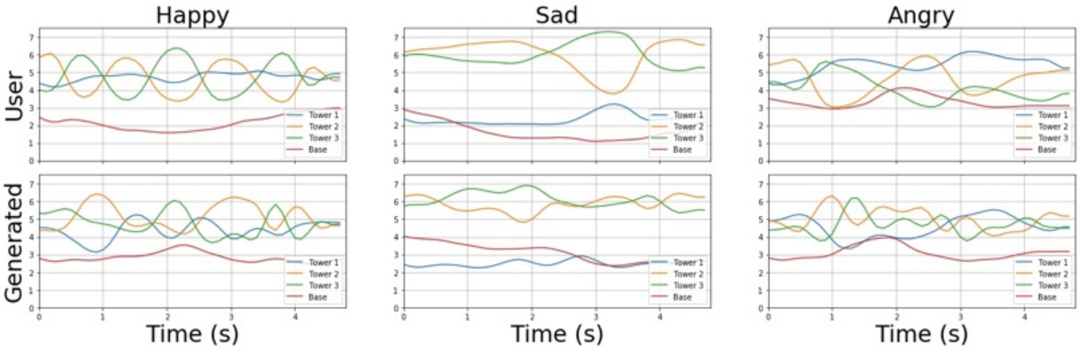


Figure 4.14: Sample trajectories of user-crafted (top) and network-generated (bottom) movements from the 3PV dataset. The generated movements retain the characteristics of the original user-crafted movements.

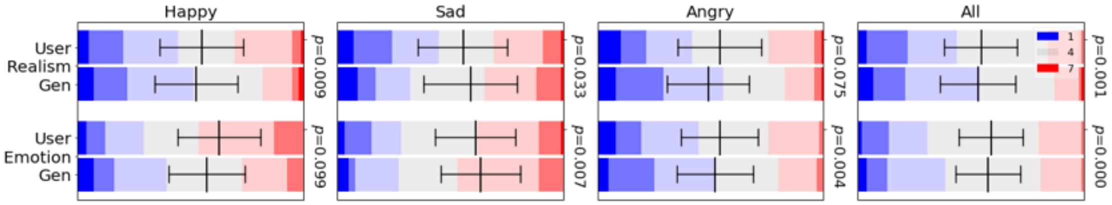


Figure 4.15: Likert scale responses from the movement comparison survey. As in Figure 4.7, color indicates level, width indicates proportion of responses for a given level, and black bars indicate means and standard deviations. For each user, the scores for each emotion (happy, sad, or angry) and source (user-crafted or generated) are calculated and rounded to the nearest integer (e.g. a given user's responses for realism for all happy user-crafted videos they saw are averaged and rounded into a single Likert score, which represents one data point used in the top left bar). p-values of **H3** tested with equivalence tests (two one-sided t-tests, equivalence bound of 0.6) are displayed on the right sides. The two sources are largely comparable, except for user-crafted happy movements being more emotive and angry movements being more realistic.

We empirically found that an embedding size of 36 was the lowest before noticeably degrading reconstruction performance. The encoder convolutions have a stride of 2 to progressively increase the effective receptive field. We trained the network for 10 epochs with a learning rate of 2×10^{-3} and a batch size of 32. We used Leaky ReLU activations ($\alpha = 0.01$), batch normalization [209], and 10% dropout after the convolutional and dense layers, as well as a mixup parameter of 0.2 [172]. For the reconstruction loss, we used mean absolute error for the front (tower 1) and base DoFs, mean squared error for the side (towers 2 and 3) DoFs, and weighed the errors as 5, 7, and 10 for the front, side, and base DoFs, respectively. For the classification loss, we used categorical cross entropy on the softmax output of the classifier. For the overall loss, we applied weights of 5 and 7 for the reconstruction and classification losses, respectively, and implemented a $\beta = 0.1$ weight for the VAE's Kullback-Leibler divergence [165].

Network training results

We trained the networks on both datasets with varying dataset sizes as an ablation study (Figure 4.11). We found that the 3PV dataset required less tuning to achieve better results. There is a noticeable improvement for the overall loss compared to using only 10% of the dataset, but only marginal improvement compared to using 50%. While it appears that smaller training datasets do not dramatically impact classification accuracy, the testing dataset sizes were also decreased; the high classification accuracies with smaller datasets are actually "overfit" and thus less generalizable to unseen samples.

Movement reconstruction results

We compared movement reconstruction accuracy with varying dataset sizes (Figure 4.12). Reconstruction fidelity increases with more data, most noticeably in the base motion. The network captures the overall trajectories but has difficulty achieving the same level of exaggeration and reconstructing granular motions, such as low-amplitude high-frequency jitter.

Embedding separability results

We used t-SNE to further compress the 36D embeddings into visualizable 2D representations (Figure 4.13). As corroborated by the classification accuracies, the emotion clusters are more separable in the 3PV dataset than the 1PV dataset. This suggests that the 3PV movements are more diverse and will yield more emotionally legible generated movements.

Movement generation results

To generate new movements, we first randomly sampled about the embedding distributions of each emotion (e.g. for a new happy movement, we sampled a 36D embedding about the mean and standard deviation of the happy embeddings), then passed these embeddings through the VAE decoder to generate full 48×4 movements. Upon inspection, the generated movements look comparable to the user-crafted movements (Figure 4.14).

Movement comparison survey results

We deployed the movement comparison survey on Prolific, offered $2 in compensation for approximately 10 minutes of work, and received 100 responses. Each user watched and rated 15 random movements out of the total set of 24 movements. We averaged each user's responses for each emotion, source, and measure, then rounded to the nearest integer on the Likert scale (e.g. a given user's responses for realism for all happy user-crafted videos they saw are averaged and rounded into a single Likert score, which represents one data point used in the top left bar of Figure 4.15). On the unrounded per-user averages,

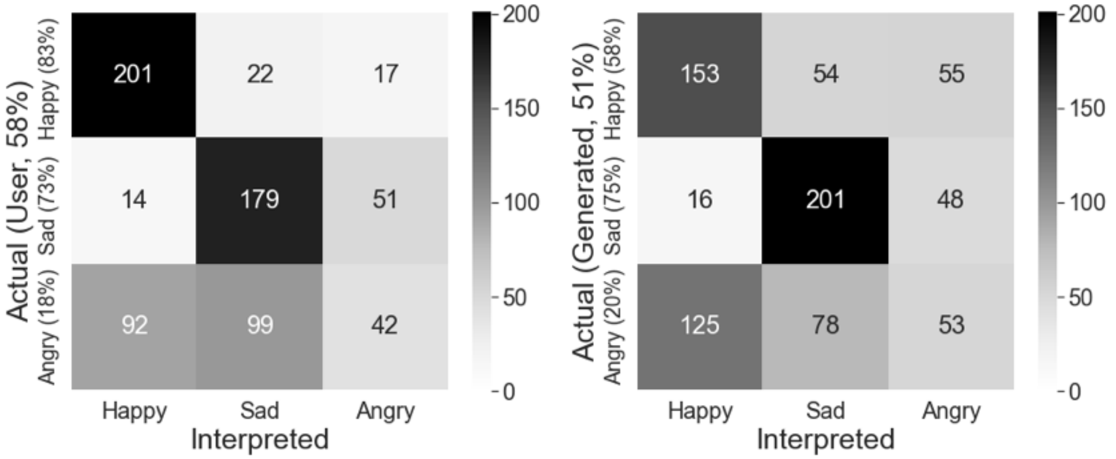


Figure 4.16: Confusion matrices for user-crafted (left) and generated movements (right) using 3PV. Overall and within-emotion accuracies accompany the vertical labels. Happiness and sadness are largely correctly matched in both sources, but anger is rarely chosen.

we used equivalence tests (two one-sided t-tests) with an equivalence bound of 0.6 (1/10$^{\text{th}}$ of the seven-point Likert scale) to test **H3**. The results show that the generated movements are comparable to the user-crafted movements in many measures, except for user-crafted happy movements being more emotive and angry movements being more realistic.

We compared the recognition rates between the actual and interpreted emotions (Figure 4.16). The recognition accuracies are well above chance (33%) for both the user-crafted and generated movements. Looking at the row-wise results, happiness and sadness are recognized with high accuracies, though generated happy movements are more ambiguous. Anger has low recognition rates in both sources, and the column-wise responses indicate that users selected anger much less frequently than the other emotions.

4.6 Discussion

The interface evaluation revealed strong preferences for 3PV, suggesting that an external perspective may be more useful for conveying affect remotely. The dataset evaluations showed that the user-crafted movements are usable as inputs to the neural network for generating new movements. The movement comparison survey supported movement generation as a valid approach for expanding a robot's behavior library.

Feedback to the interface evaluation was largely positive, with many participants commenting on the uniqueness and enjoyability of the experience. Several participants also commented on the robot's design, remarking on its cuteness and the fun factor in controlling the robot remotely. The robot's aesthetics may explain the strong preferences for being able to watch it move in 3PV.

Latency can explain the lower than expected synchronization and presence measures in 1PV. Compared to viewing the external robot in 3PV, 1PV may heighten the expectation of synchrony between motion and the video updating. Latency lands 1PV in a temporal uncanny valley, exacerbating the delay and negatively affecting the experience.

Latency can also explain the slower, smaller movements in 1PV. Although we did not view the users during the evaluation, it is reasonable to posit that 1PV employs more of the user's body as they must turn their their head to maintain a view of the video. In contrast, control in 3PV requires only hand and arm movements, which enables users to create faster and wider movements.

The neural network training results support performance increasing with more data, though our dataset is still magnitudes smaller than publicly available datasets for common modalities such as images or text. There are relatively few works in generative affective robot movements that generalize across different robot platforms and machine learning methods. Establishing standardized comparisons for generative movement algorithms is important for future research to build upon prior works; the GENEA Project is a recent development that aims to address this issue by providing common datasets for benchmarking [193].

The subjective comparisons of the user-crafted and generated movements show that they are largely comparable, but also indicate limitations of the robot's embodiment, particularly when emoting anger. The low survey responses for anger and user feedback regarding the robot's limitations, specifically its lack of appendages and difficulty in tracking finer motions, indicate that more DoFs are necessary for delineating subtleties in affect. Interestingly, the network classifier can outperform the human classifications (¿70% compared to ¡60%), suggesting that the network learns latent features that are not legible from the movement videos.

4.6.1 Limitations and Future Work

Latency

Latency is the largest bottleneck in the system, but is the hardest to mitigate. Although the latency measurements for the trip from the user's phone to the robot's host computer could reach as low as 10 ms, we cannot accurately measure the return latency between when the robot moves and when the video updates on the user's device. WebRTC benchmarks measured round-trip times from 400 ms on a cellular network down to below 100 ms on a dedicated university connection [210]. By contrast, virtual reality systems are expected to perform with latency below 50 ms, and ideally below 20 ms [211]. Future technical work could involve optimizing the underlying technologies to minimize the latency, and perhaps even freely adjust latency as a controlled variable to investigate its effects on the user experience.

Embodiment

While the simplicity of the robot's design enabled novice users to quickly learn the control scheme, it also limited its expressive capabilities to three DoFs. Several users noted feeling that many of their movements were very similar and expressed wanting arms to convey strong emotions, particularly anger. The robot's vertical translation and ear DoFs were removed to simplify the interface, but these motions may be significantly important for affording more expressiveness.

Remote evaluation paradigm

Due to the social distancing restrictions that were in place at the time of this work, we designed the interface evaluation to focus solely on the experience of the remote participant. This neglects studying the experience of a local participant interacting with the robot, and how a remote participant would use the system accordingly. A two-sided scenario may reveal favorable situations for 1PV, such as tasks requiring joint attention or communication in a real-time environment.

4.6.2 Design Implications

Research

Through this work, we gathered a dataset of affective movements from novice users, who provided usable samples after a short trial to acclimate to the system. The results of the interface evaluation suggest that 3PV is more enjoyable and useful for the movement generation task; future affective telepresence systems may benefit from this external perspective. The comparison survey results showed that these movements are still legible to other users, and support crowdsourcing and generation as viable methods for expanding a robot's given behavior library. Other researchers can adopt this accessible crowdsourcing approach for their own systems. For example, video-based pose trackers (e.g. OpenPose, VideoPose3D [212, 213]) can translate human motions into movements for humanoid robots [183], emancipating these systems from specialized motion capture environments. In the vein of RoboTurk [85], the remote control scheme could be adapted to source demonstrations for other LfD tasks such as locomotion or manipulation. Such open-access systems will require enforceable review policies to ensure the quality and usability of the samples, such as the two-survey approach with independent populations that we undertook in this work.

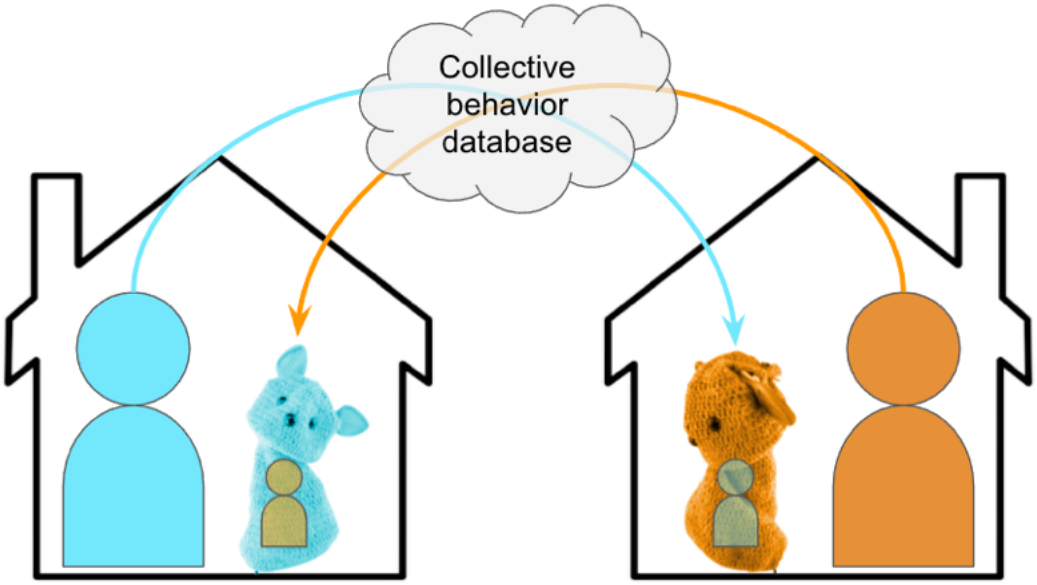


Figure 4.17: Scenario depicting remote communication through pairs of robots in separate locations. Each user remotely controls their conversation partner's robot and can record behaviors, which are stored in a personal repository on each robot and in a collective database. Coupled with behavior generation algorithms, these behaviors imbue the robot with personalities that either reflect a specific user or represent the robot as a unique individual character.

Fictional scenario

We imagine robots as a communicative medium that affords a transmission of one's physicality, adding an extra dimension beyond voice- or video-based mediums. In one example scenario[6], two family members in separate locations communicate through their conversation partner's respective robot, transmitting their voice, movement, and, optionally, their face through screens implemented on the robots. The remote users can record their movements and save them to their personal repository on their communication partner's robot. These movements are tied to a unique individual user, but are also added to a collective database of all user-crafted movements. The backend movement genera-

[6] This assumes that such social robotic systems are adopted on a similar scale as modern computing devices, either through commercial viability or open-sourcing.

tion algorithm trains on both the individual and collective samples. With the individual samples, the robot learns to act as a proxy of a specific user by generating movements in their personal idiosyncratic style. With the collective samples, the robot learns to act as a unique individual character. While movement is seemingly more innocuous than incendiary imagery or text, future work may involve safeguarding against such adversarial content.

4.7 Conclusion

We presented a variable perspective telepresence system for motion controlling a social robot and crowdsourcing affective movement samples. The system uses a smartphone as an accessible motion-based input device. Users controlled the robot from one of two perspectives: either embodying the robot from a first-person perspective through a camera in the robot's head, or a third-person perspective with an external camera looking at the whole body of the robot. To crowdsource robot movements and assess the experiential quality of the system, we performed an evaluation where lay-users created emotive movement samples for the robot. The subjective responses showed strong preferences for the third-person perspective in self-reported measures of synchronization, ease, enjoyment, engagement, and quality of the created movements. The third-person perspective also yielded movements that were faster and wider than those created in the first-person. To evaluate the usefulness of the collected dataset, we used the user-crafted movements as inputs to a neural network to generate new movements. Through a second user survey, we found that the user-crafted and generated movements were largely comparable. This work supports the use of affective telepresence systems as crowdsourcing platforms for robot demonstrations, and hopefully inspires creative approaches for conducting remote human-robot interaction research.

Part V

Conclusion

4.8 Discussion

In this section I review the work's outcomes and contributions, as well as limitations and suggestions for future work.

4.8.1 Blossom Platform

The Blossom platform is this work's primary contribution to the research community. We sought to question the conformity of social robot design (e.g. rigid, bright-colored, illuminated accents), arriving at a zoomorphic open-source design. The resulting Blossom platform features a snap-fit interior tensile actuation mechanism and soft fabric covers that are customizable by end users, even those without prior robotics experience. The movement authoring system is based on a motion-controlled smartphone application, enabling lay users to program the robot's behaviors. The artifact itself, comprising of accessible hardware and software, has been reproduced by others for their own research agendas. In the vein of research through design and knowledge embodied in the artifact, Blossom embodies several aesthetic traits – post-digital, critical design, user customization of robots – that may be useful for future HRI researchers developing their own robot platforms. In evaluating Blossom as an accessible platform, we deployed Blossoms in various contexts with lay users from diverse backgrounds, including workshops for students to create their own versions of the robot. The feedback to Blossom's design was largely positive, with many remarking that it challenged conventional notions of robot designs.

As a research platform, Blossom has several limitations and points for potential for improvement. Blossom's embodiment, though able to achieve smooth lifelike movements, is limited in its expressive range. Blossom could convincingly express happiness and sadness, but users had a particularly difficult time both creating and recognizing anger. Movement authors often expressed wanting more appendages, such as arms or a tail, to both express anger and enable more variation in the movements. Keeping Blossom's embodiment simple was a compromise in accessibility, as a more complicated embodiment would have been more difficult for lay users to control; future work would involve designing various appendage configurations. Apart from the camera used for telepresence, Blossom currently has no other sensing functionality; future configurations could explore additional sensors and output devices, such as microphones, speakers, or screen displays.

As Vandevelde noted in his thesis on the OPSORO project, developing an open-source project requires more than simply uploading the source code and design files [60]. Maintenance, documentation, and community building are key activities for open-source ecosystems. However, these activities are often dismissed as technical "plumbing" that is orthogonal to research. This misalignment deincentivizes researchers from pursuing work that does not immediately yield publishable results. Recommendations for promoting more exploratory open-source design projects include:

- Hiring technical staff specifically for supporting these projects.
- Calling on venues to reevaluate the merit of design work and technical contributions (e.g. through workshops, demonstrations, and competitions).
- Developing field-wide best-practice guides for future researchers and

roboticists, such as the Soft Robotics Toolkit [214].

4.8.2 Behavior Generation

The secondary contribution of this work is the approach for data-driven robot behavior generation. Rather than relying chiefly on expert-crafted movements programmed by roboticists, designing the movement authoring system to be accessible enabled crowdsourcing movement samples from a diverse population. We crowdsourced movement samples conveying different emotions (happiness, sadness, anger) from several volunteers. The initial dataset acts as an input for generative neural network models to create new emotive movements. The models adapted techniques from other data domains (e.g. images, text) to the new application of generative robot movement. I demonstrated two applications of this approach – movement modification and face→movement translation – and the user evaluations performed. The models produced convincing movements that could expand the robot's behavior library beyond the initial user-crafted data set.

Given the unique application of this work, there were several limitations with the approach. Though the accessibility of the system enabled crowdsourcing, the resulting dataset was magnitudes smaller than typical machine learning applications. With regards to the network architecture itself, the small dataset and uniqueness of the application constrained the network in terms of size and application. We have experimented with further applications for the network, such as implementing an embedding-based unit selection method to concatenate movements into longer sequences, inspired by prior applications in music

[215]. Another limitation, coupled to Blossom's physical design, is the limited expressive range of the network's outputs. Future work would involve scaling this approach to more complex embodiments, such as humanoid robots or virtual characters. Alternative movement authoring methods, such as pose tracking and manual mapping from humanoid to robot embodiments, could be a useful compromise between accessibility and increased expressiveness.

4.8.3 Telepresence Evaluations

The final contribution of this work is the application of Blossom as a telepresence robot for remote communication. Compared to traditional telepresence platforms which emphasize screen-based representations of the remote user, Blossom transmits physicality by employing the user's own movement in the interaction. The system could be operated in either a first-person perspective looking *through* the robot, or a third-person perspective looking *at* the robot. The user evaluations showed that the third-person perspective was preferred by users and yielded more useful movement samples for the behavior generation models.

Given the social distancing measures that were in place during the development of the telepresence work (early 2021), I was unable to perform in-person evaluations with local users interacting with the remotely controlled robot. Additionally, the overwhelming preference for the third-person perspective suggests that viewpoint affects user experience; looking *through* the robot may heighten expectations for low latency, perceptually exacerbating the network delay. A human-human interaction scenario may have elicited preferences for the first-person perspective. Other researchers are continuing to use their own Blossoms to explore other contexts, such as remote collaboration and companionship.

4.9 Conclusion

I presented an argument for using accessibility as a way to humanize the robot as a medium for communication, using the Blossom robot as an extended case study. Literal humanization of robots through humanoid appearances and human-like behaviors presents large technological challenges. However, social companion robots also pose the risk of amputating our capacity for human-human interaction. I proposed to avoid these challenges by humanizing robots through alternative designs and accessibility. Zoomorphic designs can reduce the expectations for human-like interaction, while accessibility makes their inner workings familiar to human users and frames robot development processes as opportunities for human-human interaction. I detailed three phases of Blossom's development – design, movement, and telepresence – and the efforts to make each phase accessible to non-expert users, enabling users to communicate *through* the medium of the robot. Novice users were able to contribute to each subsystem, including Blossom's physical design, behavior authoring, and application as a remote telecommunication device. Future work could address enhancing the platform's expressiveness by adding more appendages and applying the behavior generation algorithms to more complex embodiments and interaction scenarios. Given the pandemic-imposed research constraints at the time of the telepresence evaluation, studying in-person interaction through Blossom as a telepresence robot is left for future work. I hope that this project inspires a new paradigm of accessible social robot design that promotes robot-mediated communication for human-human interaction.

BIBLIOGRAPHY

[1] M. Mori, “The phenomenon of the valley of eerieness (the uncanny valley),” Energy, vol. 7, no. 4, pp. 33–35, 1970.

[2] C. E. Shannon, “A mathematical theory of communication,” The Bell System Technical Journal, vol. 27, no. 3, pp. 379–423, 1948.

[3] B. Gaver and J. Bowers, “Annotated portfolios,” Interactions, vol. 19, no. 4, pp. 40–49, 2012.

[4] J. A. Russell, “A circumplex model of affect.,” Journal of Personality and Social Psychology, vol. 39, no. 6, p. 1161, 1980.

[5] Merriam-Webster, “Robot.”

[6] K. Čapek, “Rossum’s Universal Robots,” 1920.

[7] C. L. Breazeal, Designing Sociable Robots. MIT Press, 2004.

[8] B. R. Duffy, “Anthropomorphism and the social robot,” Robotics and Autonomous Systems, vol. 42, no. 3-4, pp. 177–190, 2003.

[9] K. Dautenhahn and A. Billard, “Bringing up robots or the psychology of socially intelligent robots: From theory to implementation,” in Proceedings of the 3rd Annual Conference on Autonomous Agents, pp. 366–367, 1999.

[10] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots,” Robotics and Autonomous Systems, vol. 42, no. 3, pp. 143–166, 2003.

[11] Merriam-Webster, “Humanize.”

[12] J.-C. Giger, N. Piçarra, P. Alves-Oliveira, R. Oliveira, and P. Arriaga, “Humanization of robots: Is it really such a good idea?,” Human Behavior and Emerging Technologies, vol. 1, no. 2, pp. 111–123, 2019.

[13] K. K. Mays, Humanizing robots? The influence of appearance and status on social perceptions of robots. PhD thesis, Boston University, 2021.

[14] J. Robertson, "No place for robots: Reassessing the bukimi no tani," The Asia-Pacific Journal — Japan Focus, vol. 18, no. 4, 2020.

[15] Merriam-Webster, "Affinity."

[16] K. F. MacDorman, "Mortality salience and the uncanny valley," in 5th IEEE-RAS International Conference on Humanoid Robots, 2005., pp. 399–405, IEEE, 2005.

[17] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," IEEE Robotics & Automation Magazine, vol. 19, no. 2, pp. 98–100, 2012.

[18] E. B. Sandoval, O. Mubin, and M. Obaid, "Human robot interaction and fiction: A contradiction," in International Conference on Social Robotics, pp. 54–63, Springer, 2014.

[19] S. Kubrick, "2001: A Space Odyssey," 1968.

[20] A. Stanton, "WALL-E," 2008.

[21] Anki, "Anki vector," 2018.

[22] I. Jibo, "Jibo," 2015. Retrieved September 18, 2017 from https://www.jibo.com/.

[23] M. Robotics, "Kuri," 2018.

[24] E. Ackerman, "Consumer robotics company anki abruptly shuts down," 2019.

[25] E. Ackerman, "Jibo is probably totally dead now," 2019.

[26] E. Ackerman, "Mayfield robotics cancels kuri social home robot," 2018.

[27] A. Garland, "Ex-Machina," 2014.

[28] R. Scott, "Blade Runner," 1982.

[29] M. W. Shelley, Frankenstein; Or, the Modern Prometheus. 1818.

[30] S. Nishio, H. Ishiguro, and N. Hagita, "Geminoid: Teleoperated android of an existing person," in Humanoid robots: new developments, InTech, 2007.

[31] Hanson Robotics, "Sophia," 2016.

[32] P. Persson, J. Laaksolahti, and P. Lönnqvist, "Anthropomorphism–a multi-layered phenomenon," Proc. Socially Intelligent Agents–The Human in the Loop, AAAI Press, Technical Report FS-00-04, pp. 131–135, 2000.

[33] J. Vincent, "Sophia the robot's co-creator says the bot may not be true ai, but it is a work of art," 2017.

[34] T. Komatsu, R. Kurosawa, and S. Yamada, "How does the difference between users' expectations and perceptions about a robotic agent affect their behavior?," International Journal of Social Robotics, vol. 4, no. 2, pp. 109–116, 2012.

[35] K. Darling, The new breed: what our history with animals reveals about our future with robots. Henry Holt and Company, 2021.

[36] Merriam-Webster, "Accessibility."

[37] K. Fischer, "Tracking anthropomorphizing behavior in human-robot interaction," J. Hum.-Robot Interact., vol. 11, oct 2021.

[38] K. Cascone, "The aesthetics of failure: "post-digital" tendencies in contemporary computer music," Computer Music Journal, vol. 24, no. 4, pp. 12–18, 2000.

[39] F. Cramer, "What is 'post-digital'?," in Postdigital Aesthetics, pp. 12–26, Springer, 2015.

[40] M. Thibault, "Paper-made digital games. the poetic of cardboard from crayon physics deluxe to nintendo labo," in Proceedings of the 2018 DiGRA International Conference, 2018.

[41] V. V. Simbelis, Humanizing Technology Through Post-Digital Art. PhD thesis, KTH Royal Institute of Technology, 2018.

[42] M. McLuhan, Understanding Media: The Extensions of Man. McGraw-Hill, 1964.

[43] Merriam-Webster, "Medium."

[44] Merriam-Webster, "Communication."

[45] L. Elleström, "A medium-centered model of communication," Semiotica, vol. 2018, no. 224, pp. 269–293, 2018.

[46] J. M. Hildebrand, "What is the message of the robot medium? considering media ecology and mobilities in critical robotics research," AI & Society, pp. 1–11, 2021.

[47] S. Taipale and L. Fortunati, "Communicating with machines: robots as the next new media," Human-machine communication: rethinking communication, technology, and ourselves. Peter Lang, New York, pp. 201–220, 2018.

[48] J. F. Hoorn, "Theory of robot communication: I. the medium is the communication partner," International Journal of Humanoid Robotics, vol. 17, no. 06, p. 2050026, 2020.

[49] E. Drago, "The effect of technology on face-to-face communication," Elon Journal of Undergraduate Research in Communications, vol. 6, no. 1, 2015.

[50] S. Turkle, Alone Together: Why We Expect More from Technology and Less from Each Other. Hachette UK, 2017.

[51] C. Lacey and C. Caudwell, "Cuteness as a 'dark pattern' in home robots," in 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 374–381, 2019.

[52] C. J. Calo, N. Hunt-Bull, L. Lewis, and T. Metzler, "Ethical implications of using the Paro robot, with a focus on dementia patient care," in Workshops at the twenty-fifth AAAI conference on artificial intelligence, 2011.

[53] A. Sharkey and N. Wood, "The Paro seal robot: demeaning or enabling," in Proceedings of AISB, vol. 36, p. 2014, 2014.

[54] R. M. Williams, "I, misfit: Empty fortresses, social robots, and peculiar relations in autism research," Techné: Research in Philosophy and Technology, 2021.

[55] J. Bardzell, “Knowledge embodied in artifacts: A problem in design epistemology,” Oct 2019.

[56] M. Luria, J. Zimmerman, and J. Forlizzi, “Championing research through design in HRI,” 2019.

[57] J. Zimmerman, J. Forlizzi, and S. Evenson, “Research through design as a method for interaction design research in HCI,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’07, (New York, NY, USA), p. 493–502, Association for Computing Machinery, 2007.

[58] I. Andrews, “Post-digital aesthetics and the function of process.,” 2013.

[59] D. Richie, A tractate on Japanese aesthetics. Stone Bridge Press, 2007.

[60] C. Vandevelde, Study on the Design of DIY Social Robots. PhD thesis, Ghent University, 2017.

[61] A. Dunne and F. Raby, “Critical design FAQ.”

[62] W. G. Walter, “An imitation of life,” Scientific American, vol. 182, no. 5, pp. 42–45, 1950.

[63] W. Benjamin, “The work of art in the age of mechanical reproduction,” 1935.

[64] N. Friedman, K. Love, R. LC, J. E. Sabin, G. Hoffman, and W. Ju, What Robots Need From Clothing, p. 1345–1355. New York, NY, USA: Association for Computing Machinery, 2021.

[65] F. Thomas, O. Johnston, and F. Thomas, The Illusion of Life: Disney Animation. 1981.

[66] G. Hoffman and W. Ju, “Designing robots with movement in mind,” Journal of Human-Robot Interaction, vol. 3, no. 1, pp. 89–122, 2014.

[67] B. Wiltgen, J. M. Beer, K. McGreggor, K. Jiang, and A. Thomaz, “The interplay of context and emotion for non-anthropomorphic robots,” in 19th International Symposium in Robot and Human Interactive Communication, pp. 658–663, 2010.

[68] J. Harris, Exploring the Affect of Emotive Motion in Social Human Robot Interaction. PhD thesis, University of Calgary, 2011.

[69] R. Wistort and C. Breazeal, “Tofu: A socially expressive robot character for child interaction,” in Proceedings of the 8th International Conference on Interaction Design and Children, pp. 292–293, ACM, 2009.

[70] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” Robotics and Autonomous Systems, vol. 57, no. 5, pp. 469–483, 2009.

[71] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2013.

[72] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, “Autoencoding beyond pixels using a learned similarity metric,” 2015.

[73] A. Roberts, J. H. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” CoRR, vol. abs/1803.05428, 2018.

[74] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv Preprint arXiv:1409.0473, 2014.

[75] S. Bai and S. An, “A survey on automatic image caption generation,” Neurocomputing, vol. 311, pp. 291–304, 2018.

[76] T. Kanade, J. Cohn, and Y. Tian, “Comprehensive database for facial expression analysis,” in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), pp. 46–53, 2000.

[77] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[78] A. T. Nguyen, L. E. Richards, G. Y. Kebe, E. Raff, K. Darvish, F. Ferraro, and C. Matuszek, “Practical cross-modal manifold alignment for robotic grounded language learning,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1613–1622, June 2021.

[79] K. Goldberg, The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet. MIT Press, 2001.

[80] J. Müller, T. Langlotz, and H. Regenbrecht, "PanoVC: Pervasive telepresence using mobile phones," in 2016 IEEE International Conference on Pervasive Computing and Communications, pp. 1–10, 2016.

[81] A. E. Ainasoja, S. Pertuz, and J.-K. Kämäräinen, "Smartphone teleoperation for self-balancing telepresence robots.," in VISIGRAPP (4: VISAPP), pp. 561–568, 2019.

[82] D. Sirkin and W. Ju, "Consistency in physical and on-screen action improves perceptions of telepresence robots," in Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI '12, (New York, NY, USA), pp. 57–64, Association for Computing Machinery, 2012.

[83] K. Kilteni, R. Groten, and M. Slater, "The sense of embodiment in virtual reality," Presence: Teleoperators and Virtual Environments, vol. 21, no. 4, pp. 373–387, 2012.

[84] M. Sakashita, T. Minagawa, A. Koike, I. Suzuki, K. Kawahara, and Y. Ochiai, "You as a puppet: Evaluation of telepresence user interface for puppetry," in Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST '17, (New York, NY, USA), pp. 217–228, Association for Computing Machinery, 2017.

[85] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "RoboTurk: A crowdsourcing platform for robotic skill learning through imitation," in Proceedings of The 2nd Conference on Robot Learning, vol. 87 of Proceedings of Machine Learning Research, pp. 879–893, PMLR, 29–31 Oct 2018.

[86] S. J. Dollinger, L. Greening, and K. Lloyd, "The "mirror" and the "mask": Self-focused attention, evaluation anxiety, and the recognition of psychological implications," Bulletin of the Psychonomic Society, vol. 25, no. 3, pp. 167–170, 1987.

[87] P. Majumdar, A. Biswas, and S. Sahu, "COVID-19 pandemic and lockdown: cause of sleep disruption, depression, somatic pain, and increased screen exposure of office workers and students of india," Chronobiology International, vol. 37, no. 8, pp. 1191–1200, 2020. PMID: 32660352.

[88] M. Gržinić, "Exposure time, the aura, and telerobotics," in The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet, pp. 214–224, 2000.

[89] C. Breazeal and B. Scassellati, "How to build robots that make friends and influence people," in Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on, vol. 2, pp. 858–863, IEEE, 1999.

[90] A. Bruce, I. Nourbakhsh, and R. Simmons, "The role of expressiveness and attention in human-robot interaction," in Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, vol. 4, pp. 4138–4142, IEEE, 2002.

[91] C. F. DiSalvo, F. Gemperle, J. Forlizzi, and S. Kiesler, "All robots are not created equal: the design and perception of humanoid robot heads," in Proc of the 4th conference on designing interactive systems (DIS2002), (New York, New York, USA), pp. 321–326, ACM Press, 2002.

[92] C. C. Bennett and S. Šabanović, "Deriving minimal features for human-like facial expressions in robotic faces," International Journal of Social Robotics, vol. 6, no. 3, pp. 367–381, 2014.

[93] A. Kalegina, G. Schroeder, A. Allchin, K. Berlin, and M. Cakmak, "Characterizing the design space of rendered robot faces," in Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pp. 96–104, ACM, 2018.

[94] M. Karg, A.-A. Samadani, R. Gorbet, K. Kühnlenz, J. Hoey, and D. Kulić, "Body movements for affective expression: A survey of automatic recognition and generation," IEEE Transactions on Affective Computing, vol. 4, pp. 341–359, Oct 2013.

[95] J. Jung, S.-H. Bae, J. H. Lee, and M.-S. Kim, "Make it move: A movement design method of simple standing products based on systematic mapping of torso movements & product messages," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1279–1288, ACM, 2013.

[96] H. Tan, J. Tiab, S. Šabanović, and K. Hornbæk, "Happy moves, sad grooves: Using theories of biological motion and affect to design shape-changing interfaces," in Proceedings of the 2016 ACM Conference on Designing Interactive Systems, pp. 1282–1293, ACM, 2016.

[97] K. Baraka, A. Paiva, and M. Veloso, “Expressive lights for revealing mobile service robot state,” in *Robot 2015: Second Iberian Robotics Conference*, pp. 107–119, Springer, 2016.

[98] E. Park and J. Lee, “I am a warm robot: The effects of temperature in physical human–robot interaction,” *Robotica*, vol. 32, no. 1, pp. 133–142, 2014.

[99] Honda, “Honda 3e,” 2018.

[100] LG, “Lg exploring new commercial opportunities with expanding robot portfolio,” 2018.

[101] ASUS, “Zenbo - your smart little companion,” 2018.

[102] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, “Mechatronic Design of NAO Humanoid,” *2009 IEEE International Conference on Robotics and Automation*, no. September 2015, pp. 769–774, 2009.

[103] S. Shamsuddin, H. Yussof, L. Ismail, F. A. Hanapiah, S. Mohamed, H. A. Piah, and N. I. Zahari, “Initial response of autistic children in human-robot interaction therapy with humanoid robot nao,” in *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*, pp. 188–193, IEEE, 2012.

[104] J. Kulk, J. Welsh, *et al.*, “A low power walk for the nao robot,” in *Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA-2008), J. Kim and R. Mahony, Eds*, pp. 1–7, 2008.

[105] A. van Breemen, X. Yan, and B. Meerbeek, “icat: An animated user-interface robot with personality,” in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pp. 143–144, ACM, 2005.

[106] G. Castellano, A. Pereira, I. Leite, A. Paiva, and P. W. McOwan, “Detecting user engagement with a robot companion using task and social interaction-based features,” in *Proceedings of the 2009 international conference on Multimodal interfaces*, pp. 119–126, ACM, 2009.

[107] C. Bartneck, M. Van Der Hoek, O. Mubin, and A. Al Mahmud, “Daisy, daisy, give me your answer do!: Switching off a robot,” in *Proceedings*

of the ACM/IEEE international conference on Human-robot interaction, pp. 217–222, ACM, 2007.

[108] H. Kozima, M. P. Michalowski, and C. Nakagawa, "Keepon," International Journal of Social Robotics, vol. 1, no. 1, pp. 3–18, 2009.

[109] H. Admoni, C. Bank, J. Tan, M. Toneva, and B. Scassellati, "Robot gaze does not reflexively cue human attention," in Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 33, 2011.

[110] N. Salomons, M. van der Linden, S. Strohkorb Sebo, and B. Scassellati, "Humans conform to robots: Disambiguating trust, truth, and conformity," in Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pp. 187–195, ACM, 2018.

[111] R. Hasselvander, "Buddy: The companion robot accessible to everyone," 2017.

[112] S. Robotics, "Pepper, the humanoid robot from softbank robotics," 2014.

[113] A. Corporation, "Meet cozmo," 2017.

[114] H. Kozima, C. Nakagawa, and Y. Yasuda, "Interactive robots for communication-care: A case-study in autism therapy," in Robot and human interactive communication, 2005. ROMAN 2005. IEEE International Workshop on, pp. 341–346, IEEE, 2005.

[115] K. Wada and T. Shibata, "Living with seal robots—its sociopsychological and physiological influences on the elderly at a care house," IEEE Transactions on Robotics, vol. 23, no. 5, pp. 972–980, 2007.

[116] E. Short, K. Swift-Spong, J. Greczek, A. Ramachandran, A. Litoiu, E. C. Grigore, D. Feil-Seifer, S. Shuster, J. J. Lee, S. Huang, et al., "How to train your dragonbot: Socially assistive robots for teaching children about nutrition through play," in Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on, pp. 924–929, IEEE, 2014.

[117] H. Oh, M. D. Gross, and M. Eisenberg, "Foldmecha: Design for linkage-based paper toys," in Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, pp. 91–92, ACM, 2015.

[118] Y. Zhang, W. Gao, L. Paredes, and K. Ramani, "Cardboardizer: Creatively customize, articulate and fold 3d mesh models," in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 897–907, ACM, 2016.

[119] C. Vandevelde, F. Wyffels, B. Vanderborght, and J. Saldien, "Do-it-yourself design for social robots: An open-source hardware platform to encourage innovation," IEEE Robotics & Automation Magazine, vol. 24, no. 1, pp. 86–94, 2017.

[120] V. C. Dibia, M. Ashoori, A. Cox, and J. D. Weisz, "Tjbot: An open source diy cardboard robot for programming cognitive systems," in Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 381–384, ACM, 2017.

[121] R. Atkin, "Smartibot: The world's first a.i. enabled cardboard robot," 2018.

[122] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," Nature, vol. 521, no. 7553, p. 467, 2015.

[123] K. Goris, J. Saldien, B. Vanderborght, and D. Lefeber, "How to achieve the huggable behavior of the social robot probo? a reflection on the actuators," Mechatronics, vol. 21, no. 3, pp. 490–500, 2011.

[124] A. Singh, "Peeqo - the gif bot," 2016. Retrieved September 18, 2017 from https://imgur.com/a/ue4Ax.

[125] R. Slyper, G. Hoffman, and A. Shamir, "Mirror puppeteering: Animating toy robots in front of a webcam," in Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction, pp. 241–248, ACM, 2015.

[126] H. S. Raffle, A. J. Parkes, and H. Ishii, "Topobo: A constructive assembly system with kinetic memory," in Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 647–654, ACM, 2004.

[127] K. Nakagaki, A. Dementyev, S. Follmer, J. A. Paradiso, and H. Ishii, "Chainform: A linear integrated modular hardware system for shape changing interfaces," in Proceedings of the 29th Annual Symposium on User Interface Software and Technology, pp. 87–96, ACM, 2016.

[128] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub hu-

manoid robot: An open platform for research in embodied cognition," in Proceedings of the 8th workshop on performance metrics for intelligent systems, pp. 50–56, ACM, 2008.

[129] M. Lapeyre, P. Rouanet, J. Grizou, S. Nguyen, F. Depraetre, A. Le Falher, and P.-Y. Oudeyer, "Poppy project: Open-source fabrication of 3d printed humanoid robot for science, education and art," in Digital Intelligence 2014, p. 6, 2014.

[130] G. Langevin, "Inmoov open-source 3d printed life-size robot," 2017.

[131] ArcBotics, "Hexy the hexapod." http://arcbotics.com/products/hexy/, 2016.

[132] W. Garage, "Turtlebot," Website: Http://turtlebot. Com/Last Visited, pp. 11–25, 2011.

[133] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al., "Scratch: Programming for all," Communications of the ACM, vol. 52, no. 11, pp. 60–67, 2009.

[134] N. Fraser, "Ten things we've learned from blockly," in Blocks and Beyond Workshop (Blocks and Beyond), 2015 IEEE, pp. 49–50, IEEE, 2015.

[135] A. M. Setapen, Creating Robotic Characters for Long-Term Interaction. PhD thesis, Massachusetts Institute of Technology, 2012.

[136] J. Lasseter, "Principles of traditional animation applied to 3d computer animation," in ACM Siggraph Computer Graphics, vol. 21, pp. 35–44, ACM, 1987.

[137] T. Ribeiro and A. Paiva, "The illusion of robotic life: Principles and practices of animation for robots," in Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction, pp. 383–390, ACM, 2012.

[138] G. Hoffman, "Openwoz: A runtime-configurable wizard-of-oz framework for human-robot interaction," in 2016 AAAI Spring Symposium Series, 2016.

[139] M. Suguitan and G. Hoffman, "Blossom public repository," 2018.

[140] G. Hoffman, "Dumb robots, smart phones: A case study of music listening companionship," in RO-MAN, 2012 IEEE, pp. 358–363, IEEE, 2012.

[141] A. Kleinsmith and N. Bianchi-Berthouze, "Affective body expression perception and recognition: A survey," IEEE Transactions on Affective Computing, vol. 4, pp. 15–33, Jan 2013.

[142] M. Hashimoto, H. Kondo, and Y. Tamatsu, "Effect of emotional expression to gaze guidance using a face robot," in RO-MAN 2008 - The 17th IEEE International Symposium on Robot and Human Interactive Communication, pp. 95–100, Aug 2008.

[143] M. Poel, D. Heylen, A. Nijholt, M. Meulemans, and A. Van Breemen, "Gaze behaviour, believability, likability and the icat," Ai & Society, vol. 24, pp. 61–73, Aug 2009.

[144] M. Saerbeck and C. Bartneck, "Perception of affect elicited by robot motion," in Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction, HRI '10, (Piscataway, NJ, USA), pp. 53–60, IEEE Press, 2010.

[145] R. Desai, F. Anderson, J. Matejka, S. Coros, J. McCann, G. Fitzmaurice, and T. Grossman, "Geppetto: Enabling semantic design of expressive robot behaviors," in Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19, (New York, NY, USA), pp. 369:1–369:14, ACM, 2019.

[146] R. Gockley, A. Bruce, J. Forlizzi, M. Michalowski, A. Mundell, S. Rosenthal, B. Sellner, R. Simmons, K. Snipes, A. C. Schultz, et al., "Designing robots for long-term social interaction," in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1338–1343, Aug 2005.

[147] T. Salter, K. Dautenhahn, and R. Bockhorst, "Robots moving out of the laboratory - detecting interaction levels and human contact in noisy school environments," in RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No.04TH8759), pp. 563–568, Sep. 2004.

[148] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015.

[149] P. Ekman, "An argument for basic emotions," Cognition and Emotion, vol. 6, no. 3-4, pp. 169–200, 1992.

[150] J. Posner, J. A. Russell, and B. S. Peterson, "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology," Development and Psychopathology, vol. 17, no. 3, pp. 715–734, 2005.

[151] H. Rhodin, J. Tompkin, K. In Kim, K. Varanasi, H.-P. Seidel, and C. Theobalt, "Interactive motion mapping for real-time character control," Computer Graphics Forum, vol. 33, no. 2, pp. 273–282, 2014.

[152] Y. Seol, C. O'Sullivan, and J. Lee, "Creature features: Online motion puppetry for non-human characters," in Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '13, (New York, NY, USA), pp. 213–221, ACM, 2013.

[153] K. Yamane, Y. Ariki, and J. Hodgins, "Animating non-humanoid characters with human motion data," in Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '10, (Goslar Germany, Germany), pp. 169–178, Eurographics Association, 2010.

[154] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, "Correspondence mapping induced state and action metrics for robotic imitation," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, pp. 299–307, April 2007.

[155] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, c. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, M. Mirza, S. Jean, P.-L. Carrier, Y. Dauphin, N. Boulanger-Lewandowski, A. Aggarwal, J. Zumer, P. Lamblin, J.-P. Raymond, G. Desjardins, R. Pascanu, D. Warde-Farley, A. Torabi, A. Sharma, E. Bengio, M. Côté, K. R. Konda, and Z. Wu, "Combining modality specific deep neural networks for emotion recognition in video," in Proceedings of the 15th ACM on International Conference on Multimodal Interaction, ICMI '13, (New York, NY, USA), pp. 543–550, ACM, 2013.

[156] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI '15, (New York, NY, USA), pp. 443–449, ACM, 2015.

[157] Dongkeon Lee, Kyo-Joong Oh, and Ho-Jin Choi, "The chatbot feels you - a counseling service using emotional response generation," in 2017 IEEE

International Conference on Big Data and Smart Computing (BigComp), pp. 437–440, Feb 2017.

[158] J. M. Susskind, G. E. Hinton, J. R. Movellan, and A. K. Anderson, "Generating facial expressions with deep belief nets," in Affective Computing, InTech, 2008.

[159] I. Rodriguez, J. M. Martínez-Otzeta, I. Irigoien, and E. Lazkano, "Spontaneous talking gestures using generative adversarial networks," Robotics and Autonomous Systems, vol. 114, pp. 57–65, 2019.

[160] A. Zhou and A. D. Dragan, "Cost functions for robot motion style," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3632–3639, Oct 2018.

[161] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," Psychological Review, vol. 65, no. 6, pp. 386–408, 1958.

[162] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[163] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in 11th Annual Conference of the International Speech Communication Association, 2010.

[164] G. E. Hinton and R. S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in Proceedings of the 6th International Conference on Neural Information Processing Systems, NIPS'93, (San Francisco, CA, USA), pp. 3–10, Morgan Kaufmann Publishers Inc., 1993.

[165] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-vae: Learning basic visual concepts with a constrained variational framework.," Iclr, vol. 2, no. 5, p. 6, 2017.

[166] M. Suguitan and G. Hoffman, "Blossom: A handcrafted open-source robot," ACM Transactions on Human-Robot Interaction, vol. 8, pp. 2:1–2:27, Mar. 2019.

[167] A. Adams, M. Mahmoud, T. Baltrušaitis, and P. Robinson, "Decou-

pling facial expressions and head motions in complex emotions," in 2015 International Conference on Affective Computing and Intelligent Interaction (ACII), pp. 274–280, 2015.

[168] F. Chollet, "Keras." https://github.com/fchollet/keras, 2015.

[169] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," CoRR Abs/1609.03499, 2016.

[170] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding projector: Interactive visualization and interpretation of embeddings," 2016.

[171] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[172] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017.

[173] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," Journal of Machine Learning Research, vol. 9, no. Nov, pp. 2579–2605, 2008.

[174] S. Zhao, J. Song, and S. Ermon, "Towards deeper understanding of variational autoencoding models," arXiv Preprint arXiv:1702.08658, 2017.

[175] I. Leite, C. Martinho, A. Pereira, and A. Paiva, "As time goes by: Long-term evaluation of social presence in robotic companions," in RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication, pp. 669–674, 2009.

[176] A. Saxena, A. Khanna, and D. Gupta, "Emotion recognition and detection methods: A comprehensive survey," Journal of Artificial Intelligence and Systems, vol. 2, no. 1, pp. 53–79, 2020.

[177] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.

[178] S. Hussain, O. Ameri Sianaki, and N. Ababneh, "A survey on conversational agents/chatbots classification and design techniques," in Web,

Artificial Intelligence and Network Applications, (Cham), pp. 946–956, Springer International Publishing, 2019.

[179] S. J. Burton, A.-A. Samadani, R. Gorbet, and D. Kulić, "Laban movement analysis and affective movement generation for robots and other near-living creatures," in Dance notations and robot motion, pp. 25–48, Springer, 2016.

[180] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," ACM Transactions on Graphics (TOG), vol. 35, no. 4, p. 138, 2016.

[181] M. Marmpena, Emotional body language synthesis for humanoid robots. PhD thesis, University of Plymouth, 2021.

[182] M. Marmpena, A. Lim, T. S. Dahl, and N. Hemion, "Generating robotic emotional body language with variational autoencoders," in 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), pp. 545–551, IEEE, 2019.

[183] Y. Yoon, B. Cha, J.-H. Lee, M. Jang, J. Lee, J. Kim, and G. Lee, "Speech gesture generation from the trimodal context of text, audio, and speaker identity," ACM Transactions on Graphics, vol. 39, Nov. 2020.

[184] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, (Dublin, Ireland), pp. 69–78, Dublin City University and Association for Computational Linguistics, Aug. 2014.

[185] D. K. Jain, P. Shamsolmoali, and P. Sehdev, "Extended deep neural network for facial emotion recognition," Pattern Recognition Letters, vol. 120, pp. 69–74, 2019.

[186] N. Lubis, S. Sakti, K. Yoshino, and S. Nakamura, "Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, Apr. 2018.

[187] B. Nojavanasghari, Y. Huang, and S. Khan, "Interactive generative adversarial networks for facial expression generation in dyadic interactions," 2018.

[188] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 423–443, 2019.

[189] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021.

[190] M. Atzeni and D. R. Recupero, "Deep learning and sentiment analysis for human-robot interaction," in European Semantic Web Conference, pp. 14–18, Springer, 2018.

[191] A. Zhumekenov, M. Uteuliyeva, O. Kabdolov, R. Takhanov, Z. Assylbekov, and A. J. Castro, "Fourier neural networks: A comparative study," 2019.

[192] C. Breazeal, N. DePalma, J. Orkin, S. Chernova, and M. Jung, "Crowdsourcing human-robot interaction: New methods and system evaluation in a public environment," J. Hum.-Robot Interact., vol. 2, p. 82–111, Feb. 2013.

[193] T. Kucherenko, P. Jonell, Y. Yoon, P. Wolfert, and G. E. Henter, "A large, crowdsourced evaluation of gesture generation systems on common data: The genea challenge 2020," International Conference on Intelligent User Interfaces, 2021.

[194] M. Suguitan, R. Gomez, and G. Hoffman, "Moveae: Modifying affective robot movements using classifying variational autoencoders," in Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, pp. 481–489, 2020.

[195] R. Strong and B. Gaver, "Feather, scent and shaker: Supporting simple intimacy," in Proceedings of CSCW, vol. 96, pp. 29–30, 1996.

[196] K. Tanaka, H. Nakanishi, and H. Ishiguro, "Comparing video, avatar, and robot mediated communication: Pros and cons of embodiment," in Collaboration Technologies and Social Computing (T. Yuizono, G. Zurita, N. Baloian, T. Inoue, and H. Ogata, eds.), (Berlin, Heidelberg), pp. 96–110, Springer Berlin Heidelberg, 2014.

[197] W. D. Stiehl, J. K. Lee, C. Breazeal, M. Nalin, A. Morandi, and A. Sanna, "The huggable: A platform for research in robotic companions for pediatric care," in Proceedings of the 8th International Conference on

Interaction Design and Children, IDC '09, (New York, NY, USA), pp. 317–320, Association for Computing Machinery, 2009.

[198] R. Gomez, D. Szapiro, L. Merino, H. Brock, K. Nakamura, and S. Sabanovic, "Emoji to robomoji: Exploring affective telepresence through haru," in International Conference on Social Robotics, pp. 652–663, Springer, 2020.

[199] S. Adalgeirsson and C. Breazeal, "Mebot: A robotic platform for socially embodied telepresence," in 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 15–22, 2010.

[200] N. Tsoi, J. Connolly, E. Adéníran, A. Hansen, K. T. Pineda, T. Adamson, S. Thompson, R. Ramnauth, M. Vázquez, and B. Scassellati, "Challenges deploying robots during a pandemic: An effort to fight social isolation among children," in Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, HRI '21, (New York, NY, USA), p. 234–242, Association for Computing Machinery, 2021.

[201] A. Jonggil and G. J. Kim, "Sprint: A mixed approach to a hand-held robot interface for telepresence," International Journal of Social Robotics, vol. 10, no. 4, pp. 537–552, 2018.

[202] A. Tang, O. Fakourfar, C. Neustaedter, and S. Bateman, "Collaboration in 360° videochat: Challenges and opportunities," 2017.

[203] J. Young, T. Langlotz, M. Cook, S. Mills, and H. Regenbrecht, "Immersive telepresence and remote collaboration using mobile and wearable devices," IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 5, pp. 1908–1918, 2019.

[204] A. Denisova and P. Cairns, "First person vs. third person perspective in digital games: Do player preferences affect immersion?," in Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15, (New York, NY, USA), pp. 145–148, Association for Computing Machinery, 2015.

[205] H. Galvan Debarba, S. Bovet, R. Salomon, O. Blanke, B. Herbelin, and R. Boulic, "Characterizing first and third person viewpoints and their alternation for embodied interaction in virtual reality," PLOS ONE, vol. 12, pp. 1–19, 12 2017.

[206] G. Gorisse, O. Christmann, E. A. Amato, and S. Richir, "First- and third-person perspectives in immersive virtual environments: Presence and performance analysis of embodied users," Frontiers in Robotics and AI, vol. 4, p. 33, 2017.

[207] R. Komiyama, T. Miyaki, and J. Rekimoto, "Jackin space: Designing a seamless transition between first and third person view for effective telepresence collaborations," in Proceedings of the 8th Augmented Human International Conference, AH '17, (New York, NY, USA), Association for Computing Machinery, 2017.

[208] D. Rakita, B. Mutlu, and M. Gleicher, "Effects of onset latency and robot speed delays on mimicry-control teleoperation," in Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, HRI '20, (New York, NY, USA), pp. 519–527, Association for Computing Machinery, 2020.

[209] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the 32nd International Conference on Machine Learning (F. Bach and D. Blei, eds.), vol. 37 of Proceedings of Machine Learning Research, (Lille, France), pp. 448–456, PMLR, 07–09 Jul 2015.

[210] S. Taheri, L. A. B. Beni, A. V. Veidenbaum, A. Nicolau, R. Cammarota, J. Qiu, Q. Lu, and M. R. Haghighat, "Webrtcbench: a benchmark for performance assessment of webrtc implementations," in 2015 13th IEEE Symposium on Embedded Systems For Real-time Multimedia (ESTIMedia), pp. 1–7, 2015.

[211] K. Raaen and I. Kjellmo, "Measuring latency in virtual reality systems," in Entertainment Computing - ICEC 2015, (Cham), pp. 457–462, Springer International Publishing, 2015.

[212] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172–186, 2019.

[213] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[214] D. P. Holland, E. J. Park, P. Polygerinos, G. J. Bennett, and C. J. Walsh, “The soft robotics toolkit: Shared resources for research and design,” Soft Robotics, vol. 1, no. 3, pp. 224–230, 2014.

[215] M. Bretan, G. Weinberg, and L. Heck, “A unit selection methodology for music generation using deep neural networks,” arXiv Preprint arXiv:1612.03789, 2016.