Geppetto: Enabling Semantic Design of Expressive Robot Behaviors

Ruta Desai
Carnegie Mellon University
Pittsburgh, PA, USA
rutad@cmu.edu

Fraser Anderson
Autodesk Research
Toronto, ON, Canada
fraser.anderson@autodesk.com

Justin Matejka
Autodesk Research
Toronto, ON, Canada
justin.matejka@autodesk.com

Stelian Coros
ETH
Zurich, Switzerland
scoros@inf.ethz.ch

James McCann
Carnegie Mellon University
Pittsburgh, PA, USA
jmccann@cs.cmu.edu

George Fitzmaurice
Autodesk Research
Toronto, ON, Canada
george.fitzmaurice@autodesk.com

Tovi Grossman
Autodesk Research
Toronto, ON, Canada
tovi.grossman@autodesk.com

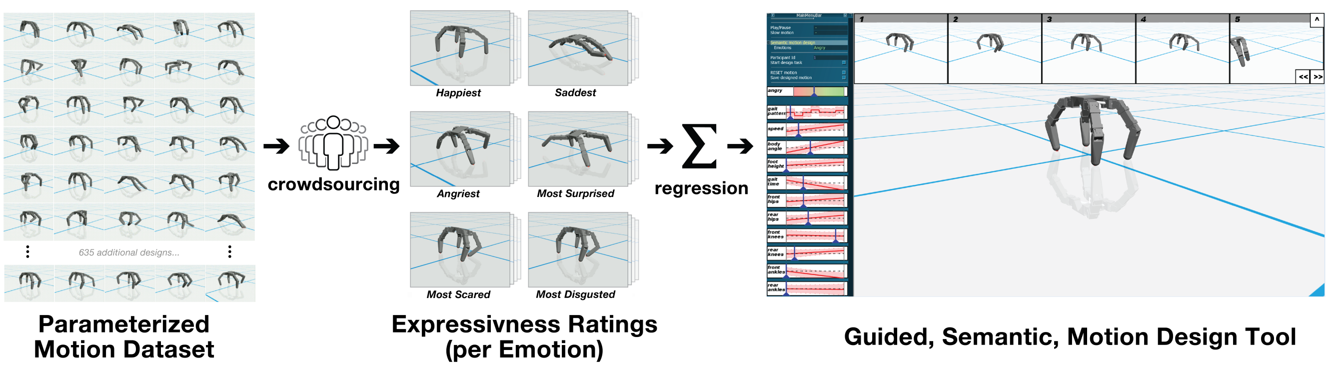

Figure 1: Overview of the semantic motion design framework– It consists of four main building blocks: (a) a dataset of parameterized expressive robot motions, (b) a crowd-powered framework for estimating the emotional perception of motions in the dataset, (c) regression analysis for establishing relationships between motion parameters and the emotional perception of the resultant motion, and (d) an intuitive design tool backed by these data-driven parameter-emotion relationships.

ABSTRACT

Expressive robots are useful in many contexts, from industrial to entertainment applications. However, designing expressive robot behaviors requires editing a large number of unintuitive control parameters. We present an interactive, data-driven system that allows editing of these complex parameters in a semantic space. Our system combines a physics-based simulation that captures the robot's motion capabilities, and a crowd-powered framework that extracts relationships between the robot's motion parameters and the desired semantic behavior. These relationships enable mixed-initiative exploration of possible robot motions. We specifically demonstrate our system in the context of designing emotionally expressive behaviors. A user-study finds the system to be useful for more quickly developing desirable robot behaviors, compared to manual parameter editing.

CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300599

CCS CONCEPTS

• **Computer systems organization** → **Robotics**; • **Human-centered computing** → *User interface design*; • **Computing methodologies** → *Interactive simulation*; *Computer graphics*.

KEYWORDS

Semantic editing; semantic design; robots; expressive robots

ACM Reference Format:

Ruta Desai, Fraser Anderson, Justin Matejka, Stelian Coros, James McCann, George Fitzmaurice, and Tovi Grossman. 2019. Geppetto: Enabling Semantic Design of Expressive Robot Behaviors. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3290605.3300599

1 INTRODUCTION

As robots become more prevalent in human environments, from factory floors to personal homes, enabling robots to express themselves can enhance and enrich our experiences and interactions with them. The paradigm of enabling robots to express intent and emotions via movements is particularly powerful [19, 34, 38, 46]. Instead of relying on anthropomorphic features or morphology, this paradigm leverages the human ability to identify emotion and intent from behavior to establish meaningful communication during interactions [1, 22, 60]. For instance, a robotic arm that collaborates with human workers on a factory floor could communicate its confusion about a task, or alert human workers if needed, by moving in a specific manner.

However, creating such expressive behaviors for robots is highly challenging [8]. Similar to digital character animation, creating behaviors for robotic characters requires tremendous skill and effort [15]. Apart from the inherent task complexity and domain knowledge requirements, robot behavior design also suffers from a lack of suitable design tools. Existing animation tools such as Blender [7] and Maya [3] enable design with absolute human control but offer limited options for integration with physical hardware. On the other hand, conventional robot control tools (e.g., ROS [42]) have extensive support for a robot's physical simulation and control, but do not allow for expressive behavior design. In comparison, our goal is to facilitate easy and intuitive design of expressive movements for robotic systems over a wide variety of applications ranging from art to social interactions.

Guided by feedback from a systematic survey of experts from animation, art, and robotics, we aim to fill this gap in existing robot behavior design tools. We present *Geppetto*, a simulation-driven robot motion design system that enables the design of expressive behaviors using high-level and semantic descriptions of behavior properties. *Geppetto* explores the creation of behaviors that convey emotions,which is an important and challenging problem within Human-Robot Interaction (HRI) [26, 52]. While in this paper the framework is used to explore emotional expression, it could easily be extended to support other semantic descriptions related to how a robot behaves, or what its movements should look like. Apart from physics-based motion simulation, *Geppetto* builds upon two recent advances in HCI and graphics research: Crowd-powered Parameter Analysis [30] and Semantic Editing [63]. These techniques are synthesized into a novel data-driven framework for the domain of robot behavior design.

Inspired by the work of Koyama et al. [30], crowdsourcing is used to obtain subjective scores pertaining to the perceptual quality of emotional expression for a generated dataset of parameterized robot motions. Using regression, functional relationships are inferred between robot motion parameters and the corresponding emotional expressions. Using these relationships, a semantic interface is developed to enable guided intuitive editing and visual exploration of the space of possible robot motions (Figure 1)[1]. A mixed-initiative approach is used for handling the unique properties of our data, e.g., the noise from crowdsourcing, and the inherent subjectivity of emotional behaviors.

This is the first system that enables casual users, without any domain knowledge of animation or robotics, to design semantically meaningful robotic behaviors. The system's utility is shown with a user-study, which indicates that users are able to create high-quality expressive robot motions. The generalizability of the framework is demonstrated by using it for two distinct robotic systems: walking robots and robotic arms.

2 RELATED WORK

This work builds upon prior work on semantic editing, crowd-powered editing, and robot motion design.

Semantic Editing and Design Space Exploration

Editing using semantic or context-specific attributes has been explored for many complex design domains such as 3D models [9, 63], images [28, 32, 44], and fonts [41]. Each of these approaches extract relevant and human-understandable attributes for their design domain, and learn a mapping between the design parameters and these attributes. With this mapping, they enable intuitive, attribute-based editing at design time. We wish to extend this methodology to the domain of robotics. Unlike the domain of 3D models and images, there is no existing large dataset of expressive robot

[1]Figures 1 and 6 of this paper have been added as looping flash animations, viewable in Adobe Acrobat. If you experience any difficulties, please see our static version of the paper, in the supplementary material.

motions. We therefore parameterize and synthesize a wide variety of such motions using a physics-based simulation.

Along with semantic editing, visual design space exploration is another useful approach. Researchers have proposed intuitive low-dimensional control spaces for predictable editing and design space exploration of complex design problems such as editing material appearance [49] or 3D models [35, 63]. Instead of finding a low-dimensional control space, we expose the current parameter space in a more visual and meaningful manner.

This work builds on Koyama et al.'s crowd-powered framework, which enabled intuitive and visual editing of continuous parameters corresponding to freely available digital content such as images [30]. Specifically, *Geppetto* deals with design spaces that consist of both continuous and discrete parameters and is particularly suited to design spaces represented by low fidelity or noisy data. *Geppetto* also allows users to combine their individual preferences with the crowd's preferences at design time.

Designing Expressive Robotic Motion

Many data-driven or model-based approaches have been explored for motion synthesis. In particular, motion capture and video data have been extensively used for increasing the style and expressiveness of anthropomorphic characters [2, 43, 50]. But it is unclear how to obtain or use such data for more generic and non-anthropomorphic robots such as robotic arms. A complementary user-driven approach is to animate toy robots or virtual characters using puppeteering [4, 11, 24, 51]. However, it is hard to pose highly articulated robots or characters to create natural looking and feasible motions using puppets. Therefore, most puppeteering based approaches are either limited to very simple characters or robots [4, 51], or they fail to account for physical feasibility [11, 24]. Similar to puppeteering are Programming by Demonstration (PbD) based approaches that enable novices to design robot motions by simple demonstrations [6]. While PbD enables easy creation of natural motions, designing semantically meaningful and expressive motions remain challenging with PbD. Finally, models that encode animation principles [58, 61] have been leveraged to improve expressiveness of robotic systems for enhanced human-robot interaction [46, 55, 56]. Unfortunately, many of these principles are abstract and generic, providing little guidance toward creating specific emotive motions from scratch. They are therefore typically used either as add-on primitives for pre-existing motions [55], or as high-level guides for manual design, similar to how animators would use them [46]. Instead, *Geppetto* enables users to design emotive behaviors by editing parameterized robot motions in simulation.

Researchers have shown a strong relation between motion parameters and attribution of affect for robots with different embodiments [25, 47]. In particular, speed and robot pose [25, 27, 57], acceleration and motion path curvature [5, 27, 47], and motion timing [27, 64] have been found to affect perceptions of motions. We therefore parameterize the walking robot's motion using features such as pose, speed, and motion timing. The robot arm's motion is parameterized in the task space instead of the joint space, inspired by how abstract trajectories could convey different emotions [5]. The system's semantically-guided parameter editing approach complements recent research on optimization-guided and keyframe-based motion editing for animated characters [29].

Crowdsourcing in Robotics and Design

Geppetto uses crowdsourcing to understand the coupled effect of various motion parameters on the overall emotional perception. Crowdsourcing enables the use of human expertise for tasks that are complex for computers, and has been widely used for a variety of tasks ranging from labeling to gathering common-sense knowledge [62]. In robotics, crowdsourcing has been used to enable robots to recognize objects or actions [21, 54], as well as for robot control [10]. Our work is most closely related to the research on understanding visual perception, and enabling better design through crowdsourcing [18, 30, 39, 63]. Our crowdsourcing pipeline is customized to deal with the greater difficulty and cost associated with evaluating motion designs, which results from the length of the motion needing to be judged, and uncertainty due to the high subjectivity of the task. Notably, we use a modified Swiss-system tournament [12] approach with an added elimination step, and use *TrueSkill* [37] to efficiently compute the perceptual quality scores for the synthesized motions.

3 CURRENT DESIGN APPROACHES

To understand the current challenges of robotic motion design, a survey of experts who design expressive behaviors for a variety of applications was conducted online.

Survey Instrument

In addition to background questions about the participants, the survey consisted of 5-pt Likert-scale and free-form questions. The questions elicited information about the types of behaviors they designed, how and why they designed them, and the time taken for the design. We also asked experts about ease of use, learnability, and suitability of the tools they used for robot behavior design tasks.

Responses

The authors reached out to a set of HRI researchers and artists/animators online to participate in the survey. The experts were selected based on their publicly available work experience information. A total of 8 experts (4 HRI researchers, 4 artists) with design experience ranging from 0.5 years - 27 years (average 11.3 years) responded to the survey. The experience of these experts covered a diverse range of contexts such as 2D/3D character behavior design, industrial and social robot design, and kinetic art sculpture.

Despite the diversity in applications, a common motivation behind designing expressive behaviors was to improve the communication, involvement, trust, and interaction of the technology they were developing (e.g., P3: *"I want my robots to be more human-readable."*, P4: *"[I want] to turn viewers into involved, emotionally invested participants"*). In response to why would they design expressive robot behaviors, experts provided further insights (P3: *"Being expressive is part of being communicative, which is critical for functional and fluent HRI. Emotion can be useful for communicating a robot's goal."*, P7: *"I see a robot's bodily motion as a lower-level means of broadcasting complex information to surrounding people."*), to tell a story and develop relationships with users (P6: *"Many engineering 'stories' do not show realistic motion which allows the viewer to dismiss the concepts."*, P2: *"To develop relationships with users through tangible actions."*).

A common theme highlighting the effort required to design behaviors also emerged. Experts who designed short-length behaviors of less than a minute (50% of our participants) reported a design time of greater than one hour. Likewise, experts who designed longer behaviors (lasting multiple minutes or hours) spent several days and sometimes several weeks on their design.

Another common theme was the lack of tools for designing robotic behaviors. Researchers and artists emphasized that the existing tools were not well suited for robot behavior design (5-pt Likert score with anchor 5 = not at all suited: M = 4.0, SD = 0.92). Experts typically relied on animation tools or ended up developing their own software. Several experts reported on the difficulty of obtaining robot simulation models (P3: *"Putting kinematic robot models into simulation takes a long time."*), pre-visualization of robot capabilities (P4: *"Pre-visualization can be quite difficult. One needs to have the actual robot working in a realistic setting in order to test it."*), and manual behavior editing (P2: *"Manually creating gestures through motor positions is tedious, unintuitive"*, P5: *"My chief problem is the lack of software tools for authoring dynamic performances with shared autonomy; I end up having to write too much software."*). Experts further reported that the tools they used were hard to learn and use (5-pt Likert score with anchor 5 = not at all easy: M = 4.12, SD = 0.64). They also emphasized the consequential challenges faced by novices in such design applications (e.g., P1: *"Having to learn lots of different, changing software and then figuring out how to connect them is difficult for beginners."*, P3: *"The toolchain is complicated, tedious."*).

Overall, the survey validated the need for improved systems for the design of expressive robot behaviors. It revealed interesting use cases and current challenges, pointing to a need for new, more intuitive, and efficient tools.

4 GEPPETTO: SEMANTIC EDITING FOR ROBOTICS

Inspired by the challenges and desires found through the survey and in the literature, *Geppetto* enables robot motion design with the help of a physics-based simulation. Parameters affecting the robot motion are presented to the user, and the system aims to reduce the domain knowledge required when modifying these parameters to create desirable motions. In particular, the system supports editing based on semantic user intent, such as designing a "happy-looking" robot. The system currently supports semantic design for six basic emotion categories – happy, sad, angry, scared, surprised, and disgusted, though it could be applied to other semantic aspects beyond emotions (e.g., expressing that a robot is busy, awaiting instruction, or friendly). The emotion categories are derived from Ekman's model of emotions [20], though we anticipate this approach can extend to other emotion models.

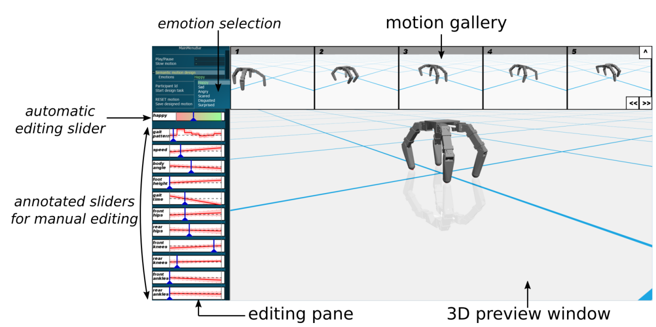

Figure 2: Geppetto's user interface.

Design Process Overview

The design process for creating an emotive robot behavior using *Geppetto* begins with users selecting a desired emotional expression (happy, sad, etc.) for the behavior from the editing pane. They can either start with a neutral default motion, or they can take advantage of the example motions in the gallery by browsing through the samples to get a sense of different motion alternatives, and then load a preferred example for further editing. Such an approach of using example-based inspiration has been found to support creativity in designers [33]. Gallery-based initialization is especially useful for novices who may not know what an expressive robot motion looks like. Once either a neutral or example motion is initialized, users can edit its expressiveness using two guided editing modes, manual and automatic. The automatic mode enables users to quickly customize the robot's motion without worrying about low-level parameter editing. Alternatively, the manual mode exposes users to parameter level editing such that they develop an understanding of which parameters create the necessary expressiveness, as well as how to edit them. With every user edit, the simulation updates the robot's motion in the preview to reflect the corresponding change.

Interface Design

The UI (Figure 2) consists of three main elements – a 3D preview window, motion gallery, and guided-editing pane. The 3D preview window renders the main robot and animates its simulated motion in real-time. The sliders in the editing pane allow users to specify the robot's motion parameters. The motion gallery displays various expressive motions of different styles for a user-specified emotion category. This gallery is populated using the emotion-specific top-ranking motions from our dataset, obtained using sampling and crowdsourcing analysis.

To design an angry robot, the user starts with the default motion, and proceeds to manually edit it using parameter sliders (Figure 3). To understand which parameters to change and how to change them, the user takes advantage of the parameter-emotion perception relationship curves visualized on each slider (Figure 4b). Based on these curves, the user increases the speed and tilts the robot's torso downwards to make it look angrier (Figure 3a). The user then leverages the example motions in the gallery for further editing. The user hovers over the preferred gallery motion to understand which parameters created it, with the help of parameter comparison cursors (Figure 4c). Inspired by the feet stomping of second gallery example, the user edits the current motion's foot height to achieve the same (Figure 3b). Finally, the user can also explore angrier motions automatically by dragging the automatic editing slider. In response, the system changes multiple parameters simultaneously to increase the motion's expressiveness. To further explore preferred motions, e.g., angrier motions with similar speed, feet stomping, and torso tilt, the user activates locking of these parameters, before dragging the automatic editing slider (Figure 3c). The system now auto-updates multiple parameters except the locked ones, to change the motion's expressiveness.

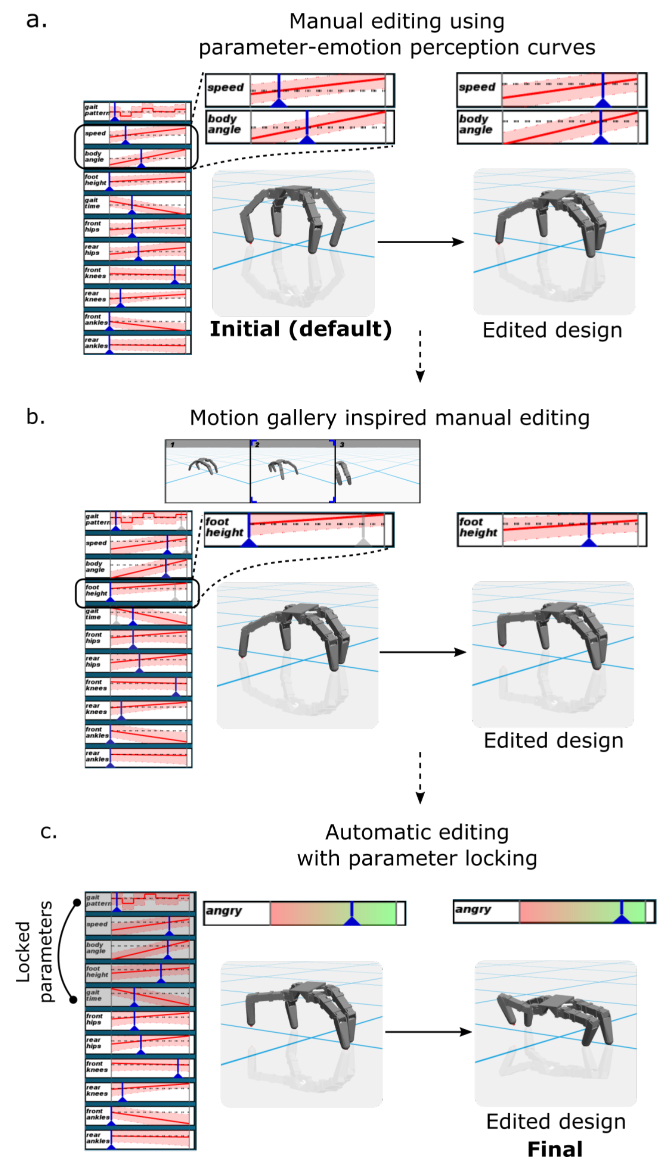

Figure 3: An example workflow designing an angry robot.

Interface Editing Features

As highlighted by the workflow, manual editing is enabled by parameter-emotion perception relationship curves and parameter comparison cursors. On the other hand, the automatic slider and parameter locking feature power automatic editing.

Parameter - emotion perception relationship curves. These curves accompany each slider, and show the effect of changing the slider's parameter on the robot's resultant emotional expression. Since these relationships are extracted from subjective crowd-sourced data, the UI also shows the system's confidence in these relationships visualized as non-linear

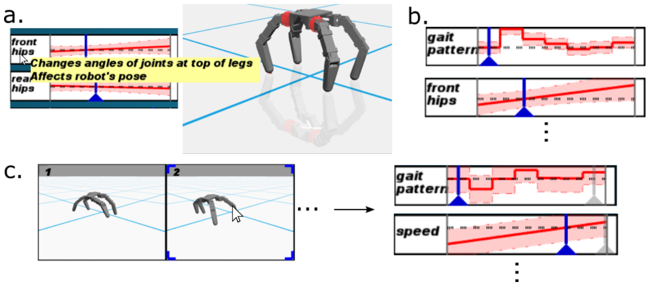

Figure 4: UI elements. (a) Parameter information is displayed as tooltips, and highlighted directly on the robot. (b) Parameter-emotion perception curve (in red) is visualized with an uncertainty band (shaded red) on each slider. The dotted line represents current motion's estimated emotional perception. (c) When a user hovers over a gallery motion, the gallery motion's parameter values are highlighted on the sliders (in light gray) alongside the current motion's parameters (blue).

error bands around the predicted score (see Figure 4b). This helps users determine the extent to which they may want to follow the curves during parameter editing. The inclusion of these error bands brings transparency to the mixed-initiative editing process, allowing the user to better collaborate with the system to achieve their goals.

Automatic editing slider. By dragging the automatic slider, users can update multiple parameters simultaneously, rather than adjusting them individually. When the position of the slider is changed, the system automatically modifies multiple parameters to achieve the corresponding change in the robot's emotional expression. This feature can be used in combination with parameter locking to achieve the desired behavior.

Parameter locking. As the automatic slider updates multiple parameters at a time, changing the automatic slider may wash out nuanced features achieved by earlier user edits. To preserve the desirable features of their current motion during automatic editing, users can lock parameters. For instance, in the example scenario of angry motion design, the user may want to maintain the speed, torso tilt, and feet stomping achieved through manual editing, while exploring better limb poses. To achieve this, the user can lock all but the pose parameters through the editing panel, and then use the automatic slider to obtain an angrier robot motion with similar speed, feet stomping, and torso tilt (Figure 4c). Note that this is much quicker than the alternative of manually editing 6 pose parameters. Parameter locking allows users to combine their design preferences with crowd-powered guidance during automatic editing. The gallery motions are also updated to show more relevant examples after parameters are locked. To update the gallery, we sort the motions in the dataset based on the similarity of parameter values locked by the user, and the quality of emotional expression. This gives users alternate motions satisfying the preferences indicated by the locked features.

Geppetto thus supports various workflows. An optimal workflow may combine both manual and automatic editing as needed. Our video shows such workflows in action.

5 IMPLEMENTATION

Motion synthesis using physics simulation

Geppetto currently supports 2 robotic platforms (Figure 5), a walking robot and an industrial robotic arm. Walking robots have been used in interactive settings such as in animatronics [16] and consumer products[53]. As a representative from the class of walking robots, we use a small quadruped robot with 3 degrees of freedom (DOF) per leg. The robotic arm is an industrial, 6 DOF KUKA arm [31]. Similar robotic arms have been used for applications requiring expressive motions such as collaborative building [59] and interactive art [23]. We consider periodic walking motions for the quadruped, and the task of moving towards a target for the robotic arm.

Motion synthesis for quadruped. The quadruped robot's motion consists of periodic coordinated limb movements (gait cycle). We leverage an existing motion generation framework that uses constrained trajectory optimization to compute valid parameter values for achieving stable walking cycles [36]. The framework achieves physical plausibility by accounting for physics-based constraints such as kinematics, gravity, friction, and hardware constraints such as motor joint and torque limits. This framework has been previously demonstrated to work on real hardware [36], and further details on these aspects can be found in the supplementary material.

Based on prior research, we expose 11 parameters affecting the robot's motion style for generating a dataset of diverse motions [25, 27, 50]. Various motion styles can be created by using different robot poses and gait patterns (e.g., galloping, trotting, walking etc.). Gait patterns are defined for a gait cycle. These patterns are characterized by the order of limb movements, and the relative phase of limb swing and stance. The robot's pose is defined using relative joint angles of the robot's limbs, as well as its global torso orientation angle defined relative to the ground plane. A pose consists of 7 angular values – torso angle, front and rear hip angles, front and rear knee angles, and front and rear ankle angles (Figure 5a). Apart from speed (1DOF), pose (7DOF), gait time (1DOF) and pattern (1DOF), we also parameterize foot height (1DOF) to create effects such as "feet stomping". The gait pattern corresponding to each gait style is discretely encoded

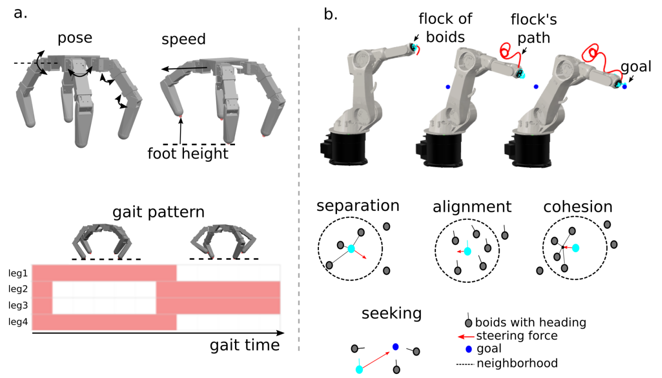

Figure 5: (a) The quadruped's motion is parameterized using robot pose, walking speed, foot height, gait time, and gait pattern (shown in red). (b) The arm is driven by a Boids flocking simulation.

using a graph (Figure 5a), while all other parameters are continuous.

Motion synthesis for robotic arm. The expressiveness of robotic arms moving towards a goal can be affected by many features, such as the curvature and smoothness of its path [5], and the variability of its speed [64]. Instead of directly prescribing the robot arm's path and speed, we use Boids flocking simulation for driving its motion – inspired by the approach used in Mimus [23].

The Boids framework uses virtual agents called boids, and a set of simple interaction rules between them to create smooth, complex and natural emergent behaviors [45]. We define a flock of m number of boids in the 3D task space, and then use the resultant average path of the flock as the target path for the robotic arm's end-effector, to be achieved through Inverse Kinematics (IK). The resultant motion and the path of the flock depend upon the interaction rules that decide each boid's movement, as a reaction to its nearby flock-mates within a small neighborhood around itself.

We use 5 interaction rules – 3 rules from the basic boids model [45]: separation, alignment, and cohesion; and 2 additional rules: goal-seeking, and exploration. Each rule creates a unique steering force that moves and updates a boid's position in the space as the simulation progresses (see Figure 5b). Separation steers a boid to avoid crowding with local flock-mates, while alignment steers it towards the average heading of the flock. Likewise, cohesion aims to move boids towards the average position of neighboring flock-mates. The seek rule complements these basic rules, by steering the boids to move towards a pre-defined goal in space (e.g., the blue goal point in Figure 5b). Finally, exploration encourages randomness in the flock by steering the boids towards a random goal intermittently. This rule is thus an extension of the seeking rule for random goals along the flock's path.

Varying the strength of each interaction rule, flock speed, and the neighborhood of influence for boids can generate diverse flocking behaviors. The strength of various rules is controlled using the corresponding weight parameters. The exploration rule is further parameterized by the sampling frequency and position of random goals. Finally, to achieve more diversity in motion generation, we also define a parameterized initialization procedure that initializes the flock to move in one of 6 specific directions for a certain period of time. In total, 11 parameters (10 continuous, 1 discrete) define the motion of the arm.

Semantic mapping framework

The semantic information about the robot motions is obtained through our mapping framework that relates the robot's motion parameter space to emotional expression space. Our framework leverages the simulation to generate a dataset of diverse motions, evaluates the emotional expression of the dataset motions using crowdsourcing, and then uses regression to obtain the mapping between motion parameters and emotional expression (Figure 1).

Motion dataset generation. We generate a dataset of diverse motions for the quadruped and the robotic arm using sampling of motion parameters. The sampling process captures the design space of possible motion styles that can be created by changing various motion parameters. We empirically choose a sampling range for all the continuous variables to generate sufficient motion variations while ensuring physical feasibility. The discrete parameters such as gait pattern for the quadruped and initial direction of boids motion for the robotic arm are uniformly sampled from a fixed set of possible values. For each sampled motion parameter set, we record an animation of the corresponding robot motion for crowdsourcing evaluation. For the quadruped, 2,000 motion parameter sets were sampled, resulting in 2,000 unique motions. 1,230 motions were physically infeasible due to collisions or instability, resulting in 670 motions for the final dataset. Similarly, 1,000 motions were sampled for the robotic arm, all of which were physically feasible and retained.

Crowdsourcing evaluation of perceived emotion. By crowdsourcing emotion perception, the system can give a relative scoring to each motion, per emotion, such that a higher score reflects a better expression of an emotion.

While there is often consensus about the particular emotion that is expressed by a motion, the degree of expressiveness is highly subjective and its perception varies between individuals. Given this, we model the score as a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ with mean μ, and uncertainty σ. To compute the score, we create a modified Swiss-system style tournament [12] where each motion sample in the dataset is treated as competitor, and competes with others to obtain the highest score per emotion category. We use the *TrueSkill* rating system [37] to convert the results of the tournament into Gaussian scores for individual samples.

To efficiently compute emotion ratings of motion samples using TrueSkill, we use an elimination-based tournament set-up instead of exhaustively competing each sample against every other. This enables us to efficiently deal with a large number of samples, and the inherent subjectivity in the data, to get the 'top' designs for each emotion. After 1 round of comparisons, wherein each sample is compared 5 times (against 5 different designs, by 5 different people), the designs ranked in the bottom half are eliminated. This process is repeated over 3 rounds (with 5, 5, and 10 comparisons), until we obtain the top motion samples for a given emotion. Elimination of ambiguous, low-ranking samples in earlier rounds allows expressive, high-ranking samples to have a higher number of comparisons against other highly-ranked designs, which improves the quality of their score estimate (reducing the corresponding uncertainty σ of the estimate). This strategy provides more accuracy for the high-ranking samples, while minimizing resources spent on ambiguous, low-ranking samples. For the quadruped dataset there were a total of 3,355 comparisons to arrive at quality rankings for the top 25% of the samples. A more naive approach of a pure round-robin without elimination would require twice the number of comparisons (6,700), and the quality of the comparisons would be lower as there would be more comparisons to low-ranking designs.

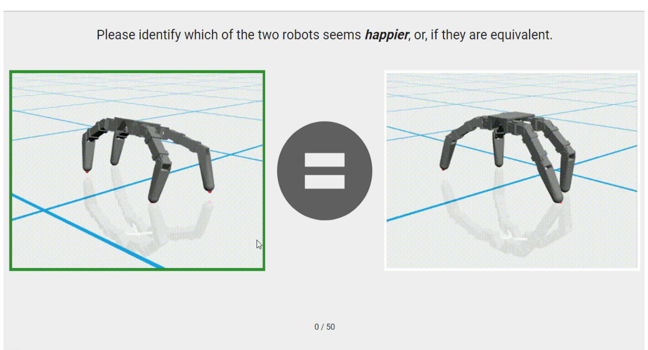

Figure 6: Interface crowd-workers used to judge emotion.

To conduct the tournament, crowd workers on Amazon Mechanical Turk serve as judges for each comparison between motion samples. For each comparison, a worker is shown a pair of robot motion videos, and asked *"Please identify which of the two robot motions seems ___, or, if they are equivalent"*, where ___ is one of: happier, sadder, angrier, more surprised, more scared, or more disgusted (Figure 6). Such a pairwise comparison approach has been preferred in the literature over asking the workers to provide an absolute score for individual samples [63]. To ensure the quality of the ratings, we enforced heuristics such as a minimum-time requirement for each comparison, and excluded data from workers who choose the same response (i.e., left, right, or equal) repeatedly (the supplementary material provides more details). In addition to crowdsourcing noise, emotion perception is subjective, so the developed interface accommodates uncertain data.

Mapping parameters to emotion. After the data is collected, a mapping between movement and perceived emotion is computed. Specifically, given an n-dimensional motion parameter set ϕ_n and a corresponding real-valued perception score μ, our goal is to learn a function $f : \phi_n \longrightarrow \mu$, that predicts the score for any seen or unseen motion represented by its parameter set. Obtaining such a function f that can estimate the perceptual quality of any emotion for a motion allows us to (a) gauge the perceptual quality of user's motion design at a given time, on the fly, and (b) help the user understand which parameters to edit, and how to edit that parameter to achieve the desired effect. The predictor function f thus powers the slider curves and the automatic slider.

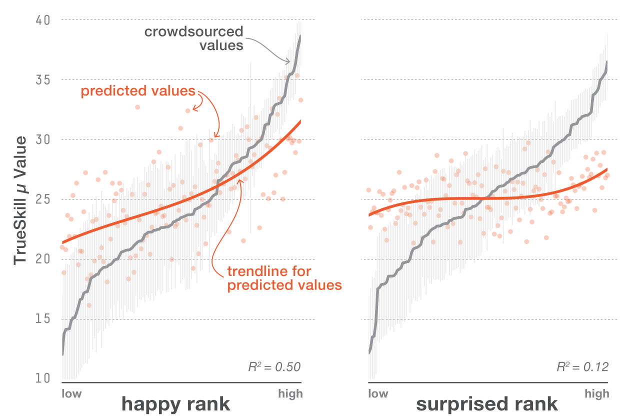

Figure 7: Comparison of predicted emotion values using linear regression (orange) with their crowdsourced values (gray) for the test samples of the quadruped motion dataset. The best (happy) and worst (surprised) fitting emotion categories are displayed.

Regression is used to construct such predictor functions that relate parameters and emotions. Both linear regression (LR) and Artificial Neural Networks (ANN) were explored for this purpose. *Geppetto* however uses LR, as LR provided a fit similar to ANN, and was much faster to execute. For the quadruped, the best-fit emotion (happy) had an R^2 score of 0.50, and the worst-fit (surprise) had $R^2 = 0.12$ (Figure 7). The variation in the fit quality for different emotion categories is an indication of the subjective nature of emotion ratings, and the inherent difficulty in expressing nuanced emotions through parameterized walking motion. To account for this, we deemphasize ambiguous and noisy samples during regression. Details about regression, and further results for all emotion categories with ANN and LR can be found in the supplementary material.

Design using predictor functions

Given a motion parameter set ϕ, a predictor function f for an emotion outputs the corresponding perception score. Let the motion parameter set corresponding to current robot motion be ϕ_n, such that it consists of n parameters p_i – $\phi_n = p_1, \ldots, p_n$. To compute the parameter-perception curve for the slider corresponding to p_1, we vary p_1 linearly over the displayed range, without changing the values of other parameters $p_2, \ldots, p_n$. The corresponding values of f are then visualized as a curve on the slider. With every manual or automatic user-initiated operation that changes the current motion parameter set ϕ_n, we dynamically update all the slider curves. The slider curves also get updated when users change the target emotion for their motion design. Since the predictor functions are obtained from noisy data, we compute and plot the 95% confidence interval (CI) for the predicted score at each point along the curve.

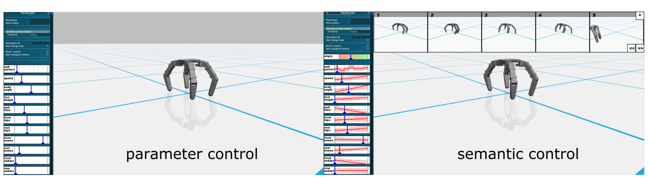

Figure 8: Interfaces used in the study's two conditions, parameter (left) and semantic (right).

The predictor functions not only predict the perceptual quality of an emotion for a motion parameter set, but also provide information about regions in the parameter space that correspond to better emotional expression. Starting from a point in the parameter space ϕ, such regions can be reached by moving along the direction of predictor function f's gradient ($\frac{\partial f}{\partial \phi}$). The automatic slider leverages this to update the robot motion. Unfortunately, since ϕ consists of both discrete and continuous parameters, we cannot compute the gradient $\frac{\partial f}{\partial \phi}$ with respect to all parameters. Consequently, when the automatic slider is used, we update the discrete and continuous parameters one by one. We first update the discrete parameter to achieve the user requested change as best as possible. Given the discrete parameter's value, we then change the continuous parameters using the gradient-based update. Specifically, for a given motion parameter set ϕ with continuous parameters set ϕ_c, the updated parameter set is $\phi'_c = \phi_c + \delta \frac{\partial f}{\partial \phi_c}$, where δ is the step-size along the gradient. The step-size is proportional to the change in the slider cursor position (Δ), which consequently reflects the desired change in robot's emotional expression (Δf). The step-size δ required to achieve the desired Δf is computed using backtracking line-search [40]. δ is positive if the user moves the automatic slider to increase the expression and is negative otherwise.

6 EVALUATION

The target audience for *Geppetto* is users without any robotics or animation background. To evaluate *Geppetto*, we therefore conducted a user-study with participants who had no experience in animation or HRI.

Participants

12 participants (9 males, 20-35 years of age) were recruited. Participants were reimbursed $25 USD for their time.

Study Design

The study had a within subject design, with participants creating expressive motions for the quadruped using two versions of the system (Figure 8). The parameter control UI allowed editing robot motion parameters with sliders but did not provide informative curves, automatic slider, or the gallery. The semantic control UI was the full interface as described above. We thus compare *Geppetto* with a diminished version of itself. This is because state-of-the-art tools (e.g., Maya) either do not account for robot's physics or would require custom plugins to do so. Further, it would be hard to test complex tools such as Maya with our target audience of casual users due to the vast amount of prior knowledge the users would need to even begin to use the tools. Since the quality of guidance provided by the semantic control UI depends upon the predictor function accuracy for an emotion, the emotion categories with highest (happy, sad), and the lowest (surprised) predictor function accuracy were used. The order of the UI conditions and emotions were counterbalanced.

Procedure

The study began with an overview of the design task for 5 minutes, followed by participant training and motion design sessions for 50 minutes, concluding with a 5-minute survey. The survey consisted of 5-pt Likert-scale questions (anchors: 1= not at all; 5 = extremely) to understand user perception of various UI features and overall design experience (see the supplementary for more details). For each condition, the participants were given the UI's demo and training for up to 10 minutes. Post training, participants were given up to 5 minutes each for designing happy, sad, and surprised robot motions. Thus, each participant designed 6 robot motions in total. Participants' motion designs were automatically saved every 30 seconds, as well as when they indicated they were finished.

Results

Quantitative results. We compare the parameter control and semantic control UI using two quantitative measures – design time and design quality. The perceptual quality of emotional expression in the user-created motion designs is evaluated using crowdsourcing, with the top and bottom 5 synthesized designs for each category included in the tournament. These top and bottom-most synthesized designs were chosen based on their prior crowdsourcing scores. The tournament structure and crowdsourcing pipeline (rounds, design comparison and elimination strategy, crowdworker filtering) are similar to our earlier crowdsourcing experiments on synthesized designs. We analyze corresponding crowdsourcing scores using confidence intervals and effect sizes, instead of null hypothesis significance testing [14]. This choice is inspired by increasing concerns over such hypothesis testing for experimental results in various research fields [13, 17, 48].

The resultant scores from crowdsourcing show that on average, users were able to create better expressive motions across all emotions using the semantic UI (Figure 9). The corresponding effect sizes (Cohen's d) between semantic vs. parameter control UI for happy, sad and surprised categories are $d = 0.79$, 0.64, and 0.35 respectively, which represent moderate-to-high effect sizes. Although surprised motions from the semantic UI scored higher on average than motions from the parameter control UI (Figure 9), the lower effectiveness (0.35) can be attributed to the limited data-driven guidance available, highlighting the effect of data certainty on semantic UI's performance.

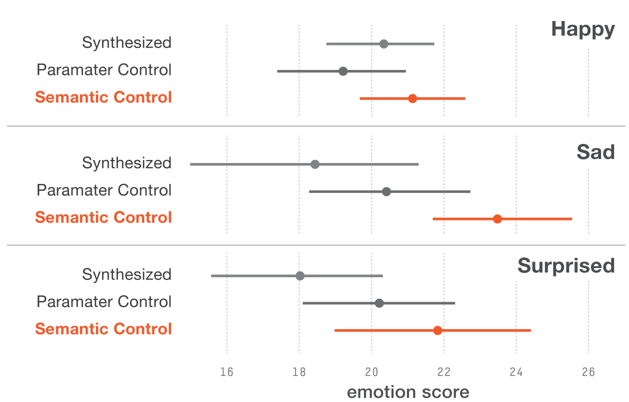

Figure 9: Mean emotion perception scores of the top 5 designs from the original dataset (Synthesized) with those created by the study participants. Bars show 95% CIs.

We also find that the designs created using the semantic UI outperform the best motions from our original dataset (Figure 9, semantic vs. synthesized). This points toward both the strengths and drawbacks of our system. The dataset synthesized using sparse random sampling may not be capturing the design space with high fidelity. Subjective crowdsourcing analysis of the dataset adds further ambiguity and noise to the data. Despite this, *Geppetto* allows users to explore beyond the synthesized dataset by enabling and leveraging their intuition of parameters at design time, guided by the emotion predictor functions.

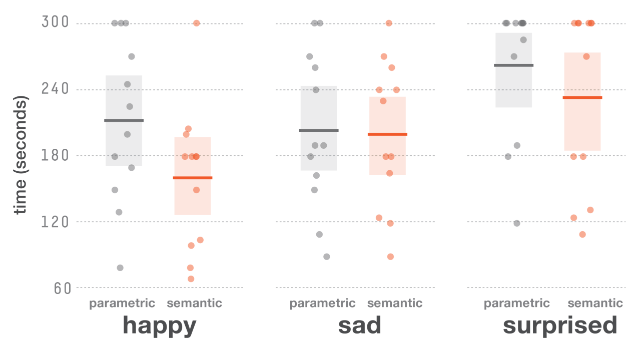

Figure 10: Individual and average design times are shown using dots and lines respectively, for both of our UIs. Shaded regions represent 95% CI.

Along with obtaining more emotive final outcomes, the participants also tended to take less design time on average with the semantic UI (Figure 10). Overall, the semantic UI enables users to start with better designs and to explore higher quality designs during their session (Figure 11). The higher initial scores of designs from the semantic UI in Figure 11 can be attributed to the use of motion gallery. After this initial boost, however, the semantic UI enables users to further improve the quality of their designs through features such as the annotated sliders and parameter comparison cursors. This is evident in the upward slope of the orange line representing semantic UI in Figure 11. Thus, the gallery, the annotated sliders, and parameter comparison cursors together provide a powerful workflow that allows users to achieve more optimal designs.

Qualitative feedback and observations. The survey provided further insights about designing with *Geppetto*. All participants reported that they are extremely likely to prefer semantic control UI to parameter control UI (5-pt Likert score: M = 4.67, SD = 0.49). Participants also believed that with the semantic control UI, they could create relatively better designs (M = 4.67, SD = 0.49), in less time (M= 4.83, SD = 0.38). This feedback further corroborates the quantitative results.

Participants' design satisfaction varied across emotions, and was related to the quality of semantic information provided. Consequently, 11 of 12 participants were satisfied with their happy design, while only 2 participants were satisfied with their surprised design.

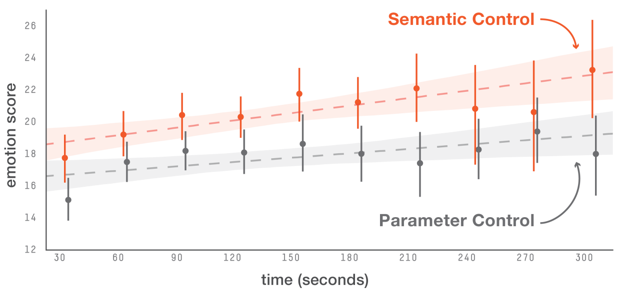

Figure 11: The evolution of the quality of user-designs (bars represent 95% CIs). The dotted lines represent the linear fit of mean scores over all emotions and participants, and the bands are a 95% CI around the fit.

The participants also provided feedback on individual UI features. 10 participants found the motion gallery and slider curves to be extremely or very useful. The parameter-comparison cursors and the automatic slider were also found to be extremely or very useful by 6 participants. The gallery catered to participants who were unclear about how to express an emotion, as well as to participants who had rough ideas about their desired design, by providing them with design alternatives. Uncertainty information on slider plots was also found to be useful. Specifically, 2 participants commented that since the surprised emotion slider curves had high uncertainty (surprise is our worst-fitted emotion), they trusted the curves less, and explored editing on their own. Parameter locking was only found to be very useful by 3 participants. Only the participants who had a clearer idea of what they wanted used parameter locking. Finally, 8 participants reported the semantic UI to be extremely or very useful in developing an understanding of the effect of parameters on emotional expressions.

The feedback and usage patterns point to the diversity of interactions and workflows that emerged during the study. Participants combined manual and automatic editing features fluidly. The feature usage also varied across participants. For instance, some subjects only used the motion gallery for design initialization, while some others leveraged it, with the help of parameter comparison cursors, to better learn and understand how specific body poses and other subtle motion features could be achieved. The automatic slider was also used in multiple ways; some used it to fine-tune their motion designs, while others used it to obtain a good starting point especially when they were dissatisfied with the gallery examples. This highlights the dependence of workflows on the noise in the data and accuracy of semantic information. Since surprise was not well captured by the gallery, participants used the automatic slider the most for this emotion.

Participants' responses also highlighted the limitations of *Geppetto*. Some participants found the automatic slider to be very aggressive since it caused major changes to the motion, meaning nuanced features, which may have existed prior, were lost. While parameter locking helps with capturing user intent about desired improvement and preserving nuances, it needs more understanding of the parameters and desired motion characteristics for effective use. The majority of participants requested an edit history and better navigation of their design trajectory. Some participants also requested the ability to edit robot structure and aesthetics for more expressiveness. Finally, participants echoed the need of capturing and enabling motion design with additional semantic information. Many participants thought about expressive motions in the space of actions and wanted to understand the mapping between parameters and space of possible and meaningful actions, so as to combine these actions into a behavior. For instance, one participant wanted to edit the parameters to make the robot drag its feet for appearing sad, while another participant wanted the robot to jump in place to express excitement. While our gallery enables users to map parameters to these desirable actions indirectly, users may or may not find the action they are looking for in the gallery.

7 GENERALIZATION, SCALABILITY, AND HARDWARE TRANSFER

Geppetto's framework generalizes across different kinds of robots as demonstrated by the quadruped and robot arm examples. Overall, the most challenging part of making *Geppetto* work for a new robot is obtaining a parameterized motion simulation. Once a dataset of motions is created, *Geppetto* requires approximately 4.5 hours and $150 USD for crowdsourcing per expression. While this may not be a significant amount of effort, re-using semantic information extracted from a particular robot's dataset to enable the design of a robot with different morphology will improve the scalability of *Geppetto*. Such transfer of the data-driven semantic map between robots, however, is dependent on the underlying motion behavior parameterization. For the quadruped, since the motion is parameterized in the joint space, the parameterization and the corresponding semantic map is dependent on the robot's morphology. To design behaviors of a six-leg hexaped, for instance, using the quadruped's semantic map, the joints of the extra pair of hexaped's legs will have to be mapped to quadruped's front or back leg joints. Since this might limit the possible hexaped behaviors that can be designed with *Geppetto*, collecting a new hexaped motion dataset might be more preferable over re-using the quadruped dataset.

On the other hand, the task space parameterization of the robot arm's motion is independent of the robot morphology. The corresponding semantic map might hence transfer more readily to different types of robot arms. We conducted a preliminary experiment to validate this. Using the semantic map of the 6 DOF KUKA arm, we tried to design motions for a 5 DOF custom robot arm. Apart from the difference in number of joints (DOF), the two robot arms also had very different link lengths. We were able to directly transfer the pre-synthesized motions of KUKA arm to the custom arm, while maintaining the resultant expression to a reasonable extent (corresponding examples can be found in the video). Further work is needed to validate this explicitly.

Geppetto's physics-based simulation supports the hardware transfer of designed motions, and has been previously verified on real hardware [36]. However, such a transfer is sometimes impeded by extreme user edits, which may drive the robot's motion parameters to infeasible regions at design time (e.g., an extreme user-commanded quadruped pose could result in the robot dragging its body on the ground – an infeasible motion for a real robot). *Geppetto* currently warns the users when such infeasibility occurs, and enables them to manually rectify such extreme edits. Finally, certain high-speed motions such as angrier robot arm motions may transfer with lower fidelity, owing to safety factors in collaborative settings with industrial robot arms.

Another factor that may affect the hardware transfer of motions from *Geppetto* is the embodiment of the physical robot. While *Geppetto* allows users to observe the simulated motion in 3D, from different viewpoints and zoom levels, it is likely that a full-scale robot may change the emotion perception from that of the simulated motion. This points to an interesting future user-study, which can specifically focus on the effect of robot's physical scale and its embodiment on emotion perception. There is also an exciting opportunity of porting *Geppetto* to a virtual reality (VR) platform, for supporting higher fidelity design.

8 DISCUSSION AND FUTURE WORK

Currently, *Geppetto* allows design space exploration and editing given a single high-level semantic goal. Enabling concurrent design to express a mixture of emotions will provide the users with greater flexibility of design. Further, high-level semantics could be coupled with task-specific mid-level semantics to better capture user-intent. For instance, mid-level semantics corresponding to emotionally expressive motion design may correspond to actions such as dragging feet, jumping, or appearing crouched. Beyond composite behaviors, extending *Geppetto* to support the design of interactive behaviors is also an interesting direction of future work.

Geppetto will also benefit from better dataset generation techniques. In particular, adaptive sampling would allow the system to capture the design space with more fidelity. On-demand sampling at design time may also enable *Geppetto* to provide guidance based on user-preferences.

Finally, any simulation-driven design system can only be as good as the underlying simulation. Our current motion parameterization and simulation do not produce motions suitable for conveying subtle emotions such as disgust and surprise. Parameterizing and synthesizing emotionally expressive robotic behaviors is an exciting future area of research. We also currently limit ourselves to the creation of robotic expressions through motions only. However, aesthetics and physical structure are equally important for visual appeal. Parameterization and editing of aesthetics is thus an interesting open problem. In particular, we envision a semantic design system that exposes the coupling of structure and motion towards creating appealing robots. Such a system will not only support the design of next generation of social and collaborative robots, but will be equally valuable for consumer robotics.

9 CONCLUSION

Towards increasing the accessibility of robot behavior design, we presented a simulation-driven and crowd-powered system that enabled semantic design of robot motions. Despite the subjectivity of the task, the system enables an intuitive design experience with the help of data-driven guidance and design space exploration, as demonstrated by our user study. We hope that our work will lead to the development of additional tools that allow both novices and experts to create desirable robots, and more broadly, open the door to future investigations of mixed-initiative interfaces across all domains of design.

REFERENCES

[1] Kaat Alaerts, Evelien Nackaerts, Pieter Meyns, Stephan P Swinnen, and Nicole Wenderoth. 2011. Action and emotion recognition from point light displays: an investigation of gender differences. *PloS one* 6, 6 (2011), e20989.

[2] Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. 2016. Modeling Stylized Character Expressions via Deep Learning. In *Asian Conference on Computer Vision*. Springer, 136–153.

[3] Autodesk 2018. *Autodesk Maya*. Autodesk. https://www.autodesk.com/products/maya/overview.

[4] Connelly Barnes, David E Jacobs, Jason Sanders, Dan B Goldman, Szymon Rusinkiewicz, Adam Finkelstein, and Maneesh Agrawala. 2008. Video puppetry: a performative interface for cutout animation. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 124.

[5] Lyn Bartram and Ai Nakatani. 2010. What makes motion meaningful? Affective properties of abstract motion. In *Image and Video Technology*

(PSIVT), 2010 Fourth Pacific-Rim Symposium on. IEEE, 468–474.

[6] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. 2008. Robot programming by demonstration. In *Springer handbook of robotics*. Springer, 1371–1394.

[7] Blender Foundation 2018. *Blender*. Blender Foundation. https://www.blender.org/.

[8] Cynthia Breazeal, Atsuo Takanishi, and Tetsunori Kobayashi. 2008. Social robots that interact with people. In *Springer handbook of robotics*. Springer, 1349–1369.

[9] Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. 2013. Attribit: content creation with semantic attributes. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 193–202.

[10] Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741* (2017).

[11] Loïc Ciccone, Martin Guay, Maurizio Nitti, and Robert W Sumner. 2017. Authoring motion cycles. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 8.

[12] László Csató. 2013. Ranking by pairwise comparisons for Swiss-system tournaments. *Central European Journal of Operations Research* 21, 4 (2013), 783–803.

[13] Geoff Cumming. 2014. The new statistics: Why and how. *Psychological science* 25, 1 (2014), 7–29.

[14] Geoff Cumming and Sue Finch. 2005. Inference by eye: confidence intervals and how to read pictures of data. *American Psychologist* 60, 2 (2005), 170.

[15] Sébastien Dalibard, Nadia Magnenat-Talmann, and Daniel Thalmann. 2012. Anthropomorphism of artificial agents: a comparative survey of expressive design and motion of virtual Characters and Social Robots. In *Workshop on Autonomous Social Robots and Virtual Humans at the 25th Annual Conference on Computer Animation and Social Agents (CASA 2012)*.

[16] Disney 2003. *Lucky the Dinosaur*. Disney. https://disneyparks.disney.go.com/blog/2013/08/today-in-disney-history-lucky-the-dinosaur-walks-on-the-scene/.

[17] Pierre Dragicevic, Fanny Chevalier, and Stephane Huot. 2014. Running an HCI experiment in multiple parallel universes. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 607–618.

[18] Abhimanyu Dubey, Nikhil Naik, Devi Parikh, Ramesh Raskar, and César A Hidalgo. 2016. Deep learning the city: Quantifying urban perception at a global scale. In *European Conference on Computer Vision*. Springer, 196–212.

[19] Magda Dubois, Josep-Arnau Claret, Luis Basañez, and Gentiane Venture. 2016. Influence of Emotional Motions in Human-Robot Interactions. In *International Symposium on Experimental Robotics*. Springer, 799–808.

[20] Paul Ekman and Wallace V Friesen. 1971. Constants across cultures in the face and emotion. *Journal of personality and social psychology* 17, 2 (1971), 124.

[21] Victor Emeli. 2012. Robot learning through social media crowdsourcing. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2332–2337.

[22] Vittorio Gallese, Christian Keysers, and Giacomo Rizzolatti. 2004. A unifying view of the basis of social cognition. *Trends in cognitive sciences* 8, 9 (2004), 396–403.

[23] Madeline Gannon. 2017. *Human-Centered Interfaces for Autonomous Fabrication Machines*. Ph.D. Dissertation. Carnegie Mellon University.

[24] Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig animation with a tangible and modular input device. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 144.

[25] John Harris and Ehud Sharlin. 2011. Exploring the affect of abstract motion in social human-robot interaction. In *RO-MAN, 2011 IEEE*. IEEE, 441–448.

[26] Malte F Jung. 2017. Affective Grounding in Human-Robot Interaction. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 263–273.

[27] Heather Knight and Reid Simmons. 2016. Laban head-motions convey robot state: A call for robot body language. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2881–2888.

[28] Adriana Kovashka, Devi Parikh, and Kristen Grauman. 2012. Whittlesearch: Image search with relative attribute feedback. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2973–2980.

[29] Yuki Koyama and Masataka Goto. 2018. OptiMo: Optimization-Guided Motion Editing for Keyframe Character Animation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 161.

[30] Yuki Koyama, Daisuke Sakamoto, and Takeo Igarashi. 2014. Crowd-powered parameter analysis for visual design exploration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 65–74.

[31] KUKA 2017. *Industrial robots*. KUKA. https://www.kuka.com/en-us/products/robotics-systems/industrial-robots.

[32] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 149.

[33] Brian Lee, Savil Srivastava, Ranjitha Kumar, Ronen Brafman, and Scott R Klemmer. 2010. Designing with interactive example galleries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2257–2266.

[34] Yueh-Hung Lin, Chia-Yang Liu, Hung-Wei Lee, Shwu-Lih Huang, and Tsai-Yen Li. 2009. Evaluating emotive character animations created with procedural animation. In *Intelligent Virtual Agents*. Springer, 308–315.

[35] Justin Matejka, Michael Glueck, Erin Bradner, Ali Hashemi, Tovi Grossman, and George Fitzmaurice. 2018. Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 369.

[36] Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, and Stelian Coros. 2015. Interactive design of 3D-printable robotic creatures. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 216.

[37] Microsoft Research 2017. *TrueSkill*. Microsoft Research. http://trueskill.org/.

[38] Brian K Mok, Stephen Yang, David Sirkin, and Wendy Ju. 2014. Empathy: interactions with emotive robotic drawers. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, 250–251.

[39] Nikhil Naik, Jade Philipoom, Ramesh Raskar, and César Hidalgo. 2014. Streetscore-predicting the perceived safety of one million streetscapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 779–785.

[40] Jorge Nocedal and Stephen J Wright. 2006. Numerical optimization. (2006).

[41] Peter O'Donovan, Jānis Lībeks, Aseem Agarwala, and Aaron Hertzmann. 2014. Exploratory font selection using crowdsourced attributes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 92.

[42] Open Source Robotics Foundation 2018. *ROS*. Open Source Robotics Foundation. http://www.ros.org/.

[43] Wei Pan and Lorenzo Torresani. 2009. Unsupervised hierarchical modeling of locomotion styles. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 785–792.

[44] Devi Parikh and Kristen Grauman. 2011. Relative attributes. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 503–510.

[45] Craig W Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics* 21, 4 (1987), 25–34.

[46] Tiago Ribeiro and Ana Paiva. 2012. The illusion of robotic life: principles and practices of animation for robots. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*. IEEE, 383–390.

[47] Martin Saerbeck and Christoph Bartneck. 2010. Perception of affect elicited by robot motion. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*. IEEE, 53–60.

[48] Frank L Schmidt. 2013. Eight Common But False Objections to the Discontinuation of Significance Testing in the. *What if there were no significance tests?* (2013), 37.

[49] Ana Serrano, Diego Gutierrez, Karol Myszkowski, Hans-Peter Seidel, and Belen Masia. 2016. An intuitive control space for material appearance. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 186.

[50] Ari Shapiro, Yong Cao, and Petros Faloutsos. 2006. Style components. In *Proceedings of Graphics Interface 2006*. Canadian Information Processing Society, 33–39.

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

[52] Sichao Song and Seiji Yamada. 2017. Expressing emotions through color, sound, and vibration with an appearance-constrained social robot. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2–11.

[53] Sony 2017. *Aibo*. Sony. http://www.sony-aibo.com/.

[54] Yuyin Sun and Dieter Fox. 2016. Neol: Toward never-ending object learning for robots. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 1621–1627.

[55] Daniel Szafir, Bilge Mutlu, and Terrence Fong. 2014. Communication of intent in assistive free flyers. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, 358–365.

[56] Leila Takayama, Doug Dooley, and Wendy Ju. 2011. Expressing thought: improving robot readability with animation principles. In *Proceedings of the 6th international conference on Human-robot interaction*. ACM, 69–76.

[57] Haodan Tan, John Tiab, Selma Šabanović, and Kasper Hornbæk. 2016. 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*. ACM, 1282–1293.

[58] Frank Thomas, Ollie Johnston, and Frank. Thomas. 1995. *The illusion of life: Disney animation*. Hyperion New York.

[59] Lauren Vasey, Tovi Grossman, Heather Kerrick, and Danil Nagy. 2016. The hive: a human and robot collaborative building process. In *ACM SIGGRAPH 2016 Talks*. ACM, 83.

[60] Gentiane Venture, Hideki Kadone, Tianxiang Zhang, Julie Grèzes, Alain Berthoz, and Halim Hicheur. 2014. Recognizing emotions conveyed by human gait. *International Journal of Social Robotics* 6, 4 (2014), 621–632.

[61] Jue Wang, Steven M Drucker, Maneesh Agrawala, and Michael F Cohen. 2006. The cartoon animation filter. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 1169–1173.

[62] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE, 766–773.

[63] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. 2015. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 86.

[64] Allan Zhou, Dylan Hadfield-Menell, Anusha Nagabandi, and Anca D Dragan. 2017. Expressive robot motion timing. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 22–31.