Geppetto: Enabling Semantic Design of Expressive Robot Behaviours

Ruta Desai¹, Fraser Anderson², Justin Matejka², Stelian Coros¹, James McCann¹, George Fitzmaurice², Tovi Grossman²

¹Carnegie Mellon University, Pittsburgh, PA, USA. {rutad, scoros, jmccann}@andrew.cmu.edu
²Autodesk Research, Toronto, ON, Canada. {firstname.lastname}@autodesk.com

ABSTRACT
We present an interactive, data-driven system for designing expressive robot behaviors. The primary goal is to increase the accessibility of robot behavior design for a variety of applications ranging from art to entertainment. The system enables robot behavior design using high-level and semantic descriptions of behavior properties such as specifying the desired emotion expression. To achieve such designs, the 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. By leveraging these relationships for a mixed-initiative design, the system guides users to explore the space of possible robot motions. A user-study finds the system to be useful for more quickly developing desirable robot behaviors.

Author Keywords
Semantic editing; semantic design; robots; expressive robots

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

INTRODUCTION
As robots become more prevalent in social environments, from factory floors to personal homes, enabling robots to express themselves can enhance and enrich our experience of interactions with them. The paradigm of enabling robots to express intent and emotions via movements is particularly powerful [12, 27, 30, 36]. Instead of relying on anthropomorphistic features or morphology, this paradigm leverages the human ability to identify emotion and intent merely from behavior to establish meaningful communication during interactions [1, 15, 46]. For instance, a robotic manipulator arm that collaborates with fellow workers on a factory floor could communicate its confusion about a task, or appear tired when it detects wear and tear, by moving in a specific manner.

However, creating such expressive movement behaviors for robots is highly challenging [5]. Similar to digital character animation, creating behaviors for robotic characters requires tremendous skill, and effort [10]. Apart from the inherent task complexity and domain knowledge requirements, robot behavior design also suffers from the lack of suitable design tools. Existing animation tools such as Blender [14] and Maya [52] 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 [53]) have extensive support for robot’s physical simulation and control, but do not allow for mixed-initiative 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 attempt to fill in this gap present 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 such as emotion conveyed by the behavior. Apart from physics-based motion simulation, Geppetto builds upon two recent advances in HCI and graphics research - Crowd-powered Parameter Analysis [23] and Semantic Editing [50]. These techniques are combined into a novel data-driven framework for the domain of robot behavior design.

Inspired by the work of Koyama et al. [23], 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 analysis, functional relationships are inferred between robot motion parameters and the corresponding emotional expressions. Using these relationships, a semantic interface is developed to enable gently guided intuitive editing, and visual exploration of the space of possible robot motions (Figure 1, right). A Mixed-initiative approach is used for handling the unique properties of our data, such as the noise from crowdsourcing, and the inherent subjectivity of emotional behaviours.

To the best of our knowledge, 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 indicated that users were able to create high-quality expressive robot motions. The generalizability of the presented framework is demonstrated by using it for two distinct robotic systems: walking robots, and manipulator arms (Figure 2).

**Figure 2:** Users can design expressive motions for two distinct types of robots: (a) a quadruped, and (b) a robotic arm, while exploring the space of possible motions.

**RELATED WORK**
This work is inspired and 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 [6, 50], images [21, 24, 34], and fonts [32]. 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 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 [38], or 3D models [28, 50]. 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. which enables intuitive editing of continuous parameters corresponding to digital content such as images, visually, using a crowd-powered framework [23]. Parameter sliders with heat-map visualizations are used to gently guide the users to a relevant region in the design space. 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. We leverage mixed-initiative design for scenarios where the available datasets capture the design space in a limited manner, or when the data is relatively noisy. This is achieved by providing relevant guidance in a transparent manner. Specifically, parameter sliders are annotated with curves that indicate both parameter-semantic attribute relationships, and the degrees of uncertainty within those relationships. Finally, unlike most crowd-powered systems, Geppetto provide user interface features that enable users to combine their individual preferences with 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, 33, 39]. However, 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 [3, 8, 17, 40]. 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 [3, 40], or they fail to account for physical feasibility [8, 17]. Finally, models that encode animation principles [45, 48] have been leveraged to improve expressiveness of...
Crowdsourcing in Robotics and Design

Crowdsourcing is used 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 and opinions [49]. In robotics, crowdsourcing has been used to enable robots to recognize objects or actions [13, 41], as well as for robot control [7]. Our work is most closely related to the research on understanding visual perception, and enabling better design through crowdsourcing [11, 23, 31, 50]. We build on these approaches and use a crowd-powered pairwise comparison approach for evaluating motions. The crowdsourcing pipeline is customized to deal with the greater difficulty and cost associated with evaluating our motion designs, which results from the length of the animation needing to be judged, and uncertainty due to the high subjectivity of the task. Notably, we use a modified Swiss-system tournament [9] approach with an added elimination step, and use TrueSkill [19] to efficiently compute the perceptual quality scores for the synthesized motions.

CURRENT DESIGN APPROACHES

To understand the current challenges of robotic motion design, a survey of experts who design expressive behaviors for applications ranging from art and entertainment to Human-Robot Interaction (HRI) was conducted.

Survey Instrument

HRI and robotics researchers, artists, and animators participated in a survey. In addition to background questions, the survey consisted of 5-point Likert-scale and free-form questions. The questions elicited information about the types of behaviors they designed, how and why they designed them, as well as the time taken and tools used in design. We also asked their opinion about the tools they used, in terms of ease of use, learnability, and suitability for various robot behavior design tasks.

Responses

Eight experts (4 HRI researchers, 4 artists/animators) with design experience ranging from 0.5 years – 27 years (average 11.3 years) participated in 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 robots was to improve the communication, involvement, 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”). Another common theme highlighted the effort required to design behaviours. 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 for their design.

Another common theme was the lack of tools for designing robotic behaviors. Researchers as well as artists emphasized that the existing tools were not well suited for robot behavior design (with an average score of 4). 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 (mean of 4.12). 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 people just starting out.”, P3: “The toolchain is complicated and tedious.”).

To understand possibilities for assisting robotic designers, we asked why they design expressive robot behaviors. Experts replied with a variety of reasons such as increasing

---

1 Task space is the lower-dimensional subspace of motion directly relevant for the task.
communication and trust of robots (P3: “Being expressive is part of being communicative, which is critical for functional and fluent human-robot interactions. 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.”).

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.

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 such semantic design for six basic emotion categories – happy, sad, angry, scared, surprised, and disgusted, though it could be extended to other semantic attributes.

Interface Design
The UI (Figure 3) 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.

![Figure 3: User interface overview. The 3D preview window renders the robot's motion. The gallery and annotated sliders provide semantically relevant information at design time.](image)

Design Process Overview
The design process for an emotive robot behavior begins with a user 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 [26]. Gallery-based initialization is especially useful for novices who may not know what an expressive robot motion looks like. Once a motion is initialized, users can edit motion's expressiveness as desired using two guided editing modes – manual, and automatic. Each mode focuses on two different, and critical requirements of casual users – fast prototyping customization, and learning. The automatic mode enables users to quickly customize the robot's motion without worrying about low-level parameter editing. The manual mode, on the other hand, exposes users to parameter level editing such that they develop an inherent 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.

![Figure 4: An example workflow designing an angry robot.](image)
To design an angry robot, the user starts with the default motion, and proceeds to manually edit it using parameter sliders (Figure 4). To understand which parameters to change and how to change them, the user takes advantage of parameter-emotion perception relationship curves visualized on each slider (Figure 5b). Based on these curves, the user increases the speed and tilts the robot’s torso downwards to make it look angrier (Figure 4a). 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 5c). Inspired by the feet stomping of second gallery example, the user edits the current motion’s feet height to achieve the same (Figure 4b). Finally, to explore angrier motions with similar speed, feet stomping, and torso tilt, the user activates locking of these parameters, and drags the automatic editing slider (Figure 4c). In response, the system changes multiple parameters except the locked ones, to increase the motion’s expressiveness.

**Interface Editing Features**

As highlighted by the workflow, manual editing leads to user understanding of parameters, and is enabled by parameter-emotion perception relationship curves and parameter comparison cursors. On the other hand, automatic editing allows quick updates of the user’s motion design, based on user’s high-level intent of increasing or decreasing the intensity of robot’s emotional expression. It is powered by automatic update slider and parameter locking feature.

*Parameter-emotion perception relationship curves*

These curves, which are accompanied with each slider, 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 error bands around the predicted score (see Figure 5b). This allows users to 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.

*Parameter comparison cursors with motion gallery*

Different sets of parameter values result in widely diverse motions; each corresponding to a different style or intensity of an emotion. To enable users in understanding how different parameter values result in motion diversity, we leverage the diverse examples in the motion gallery. Users can visualize parameter values corresponding to any motion in the gallery on the sliders, by hovering over that motion (Figure 4c). This enables the users to make parameter-level comparisons between the motions in the gallery and their design, and copy the preferred individual parameter values.

*Automatic update slider*

By dragging the automatic update 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 (below) to achieve the desired behavior.

![Figure 5: 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 corresponds to 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).](image)

*Parameter locking*

As the automatic slider updates multiple parameters at a time, changing the automatic slider by a small amount may drastically modify the resulting motion. As a result, the nuanced features of the robot’s motion achieved by a user’s earlier edits may be lost when using the automatic slider. 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 update slider to obtain an angrier robot motion with similar speed, feet stomping, and torso tilt (Figure 5c). Note that this is much quicker than the alternative of manually editing 6 pose parameters. Parameter locking thus allows users to combine their design preferences with crowd-powered guidance during automatic editing of designs.

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 parameters to the values locked by the user, and the quality of emotion expression. This gives users alternate motions satisfying the preferences indicated by the locked features.

Our system supports various workflows for motion editing. An optimal workflow could combine both manual and automatic editing as needed. Our video shows such workflows in action.
parameters are continuous.
encoded using a graph (see Figure 6a), while all other
gait pattern corresponding to each gait style is discretely
the path of the flock depend upon the
path for the robotic arm’s end-effector, to be achieved
then use the resultant average path of the flock as the target
define a flock of
smooth, complex and natural emergent behaviors [16]. We
set of simple interaction rules between them to create
The Boids framework uses virtual agents called boids, and a
motion through constrained trajectory optimization [29].
To enable the design and dataset generation of diverse
motion styles, we expose eleven parameters of the
optimization that affect the robot’s motion style. 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, and are characterized by
the order of limb movements, relative phase of limb swing,
and stance. Pose is defined using relative joint angles of
robot’s limbs, as well as its global torso orientation angle
defined relative to the ground plane. 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
6a). Apart from speed (1DOF), pose (7DOF), gait time
(1DOF) and pattern (1DOF), we also parameterize foot
height (1DOF) to create the effect of feet “stomping”. The
gait pattern corresponding to each gait style is discretely
encoded using a graph (see Figure 6a), while all other
parameters are continuous.
Motion synthesis for robotic arm
Expressiveness of robotic arms moving towards a goal can be
affected by many features, such as the curvature of its path
[4], the variability of its speed [51], and path smoothness.
Instead of directly prescribing the robot arm’s path and
speed, we use a Boids simulation to drive its motion – similar
to the approach used in Mimus. [16].
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 [16]. 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 five interaction rules -- three rules from the basic
boids model [35]: separation, alignment, and cohesion; and
two other custom 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 6b). 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 6b). 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.
Motion synthesis for quadruped
The quadruped robot’s motion consists of periodic
coordinated limb movements (gait cycle). We obtain such a
motion through physics simulation [29].
IMPLEMENTATION
Motion synthesis using physics simulation
Our system currently supports two robotic platforms (Figure
2), a quadruped walking robot, and an industrial robotic arm.
Walking robots have been used in interactive settings such as
in animatronics [54], and consumer products [55]. As a
representative from this class of robotic systems, we use a
small quadruped robot with three degrees of freedom (DOF)
per leg. The robotic arm is an industrial, six DOF KUKA arm
[56]. Similar robotic arms have been used for applications
requiring expressive motions such as collaborative building
[25], and interactive art [16]. We consider periodic walking
motions for the quadruped, while for the robotic arm, we
consider the task of moving towards a target point.
Motion synthesis for quadruped
The quadruped robot’s motion consists of periodic
coordinated limb movements (gait cycle). We obtain such a
motion through constrained trajectory optimization [29].
Diverse flocking behaviors can be generated by varying the
strength of each interaction rule, flock speed, and the
neighborhood of influence for boids. We control the
strengths of various rules using 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 six 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.

To efficiently compute emotion ratings of motion samples, we use the TrueSkill rating system [19] to convert the results of the tournament into Gaussian score estimates for individual samples.

Mapping parameters to emotion

After data is collected, a mapping between movement and perceived emotion is computed. Specifically, given an n-dimensional motion parameter set φn, and a corresponding real-valued perception score µ, our goal is to learn a function f : φn → µ, 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 to achieve the desired effect. The predictor function (f) thus powers our parameter-perception curves for manual editing, as well as our automatic update slider.

To generate the parameter emotion curves rendered on the sliders, regression was used to construct the relationship between parameter and emotion. Both linear regression and Artificial Neural Networks were explored to provide this mapping, however, linear regression provided a similar fit and was much faster to execute. For the quadruped, the best-fitted emotion (happy) had R² score of 0.50, and the worst-fitted emotion (surprise) had R² = 0.12 (Figure 8). 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 in a parameterized quadruped walking robot motion. Detailed results for all emotion categories with ANN as well as linear regression can be found in the supplementary material.

**Design using Predictor Functions**

Given a motion parameter set φn, a predictor function f for an emotion outputs the corresponding perception score. Let the motion parameter set corresponding to current robot motion be φn, such that it consists of n parameters p_i → φ_n = {p_1, ...,
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 \( \phi_n \), 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.

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 \( \phi \), such regions can be reached by moving along the direction of predictor function \( f \)'s gradient \( (df/d\phi) \). The automatic update slider leverages this to update the robot motion. Unfortunately, since \( \phi \) consists of both discrete and continuous parameters, we cannot compute the gradient \( df/d\phi \) 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 \( \phi \) with continuous parameters set \( \phi_c \), the updated parameter set \( \phi_c' \) is \( \phi_c' = \phi_c + \delta df/d\phi_c \), where \( \delta \) is the step-size along the gradient. The step-size is proportional to the change in the slider cursor position \( (\Delta) \), which consequently reflects the desired change in robot’s emotional expression \( (\Delta f) \). The step-size \( \delta \) required to achieve the desired \( df \) is computed using backtracking line-search [47]. \( \delta \) is positive if the user moves the automatic slider to increase the emotional expression, and is negative otherwise.

**EVALUATION**

To evaluate Geppetto’s efficacy and features, we conducted a user-study with participants who had no prior experience in character 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 sliders, or the gallery. The semantic control UI was the full interface as described above. 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. For each condition, the participants were given a demo of the system, and could train for up to 5 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 complete.

**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 are evaluated using crowdsourcing, with the top and bottom 5 synthesized designs for each category included in the tournament. The resultant scores show that users were able to create better expressive motions on average using the semantic UI, across all emotions (Figure 9).
We also find that the designs created using the semantic UI outperform the best motions from our original dataset in expression of various emotions (Figure 9, Semantic vs. Synthesized). This points towards 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.

The semantic UI was more efficient, as participants tended to take less time on average using the semantic UI, despite ending up with more emotive final outcomes (Figure 10). Overall, the semantic UI enables users to start with better designs and to explore higher quality designs during their session (Figure 11).

We also asked the participants for feedback on individual UI features using Likert-scale questions. 10 participants found the motion gallery and slider curves to be extremely or very useful. The parameter-comparison cursors and automatic update 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 crude ideas about their desired design by providing them with design alternatives. Uncertainty information on slider plots was also found useful. Specifically, two participants commented that since the surprised emotion parameter-emotion perception 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. Overall, the participants explored more while designing with semantic control UI owing to the availability of more features and design alternatives.

The feedback and usage patterns points 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 update slider was also used in multiple ways; some used it to fine-tune their manually edited motions, 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 our system, all participants preferred semantic control UI to parameter control UI overall (4.67 average score on 5-pt Likert scale; 5 is the highest). Participants believed that they could create relatively better designs (4.67 average), in less time with the semantic control UI (4.83 average). This feedback further corroborates the quantitative results. Participants’ design satisfaction varied across emotions, and was a function of the quality of semantic information provided. Consequently, 11 of 12 participants were satisfied with their happy design, while only 2 of 12 participants were satisfied with their surprised design.

The feedback and usage patterns points 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 update slider was also used in multiple ways; some used it to fine-tune their manually edited motions, 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 our system, all participants preferred semantic control UI to parameter control UI overall (4.67 average score on 5-pt Likert scale; 5 is the highest). Participants believed that they could create relatively better designs (4.67 average), in less time with the semantic control UI (4.83 average). This feedback further corroborates the quantitative results. Participants’ design satisfaction varied across emotions, and was a function of the quality of semantic information provided. Consequently, 11 of 12 participants were satisfied with their happy design, while only 2 of 12 participants were satisfied with their surprised design.

The feedback and usage patterns points 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 update slider was also used in multiple ways; some used it to fine-tune their manually edited motions, 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 our system, all participants preferred semantic control UI to parameter control UI overall (4.67 average score on 5-pt Likert scale; 5 is the highest). Participants believed that they could create relatively better designs (4.67 average), in less time with the semantic control UI (4.83 average). This feedback further corroborates the quantitative results. Participants’ design satisfaction varied across emotions, and was a function of the quality of semantic information provided. Consequently, 11 of 12 participants were satisfied with their happy design, while only 2 of 12 participants were satisfied with their surprised design.

The feedback and usage patterns points 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 update slider was also used in multiple ways; some used it to fine-tune their manually edited motions, 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 our system, all participants preferred semantic control UI to parameter control UI overall (4.67 average score on 5-pt Likert scale; 5 is the highest). Participants believed that they could create relatively better designs (4.67 average), in less time with the semantic control UI (4.83 average). This feedback further corroborates the quantitative results. Participants’ design satisfaction varied across emotions, and was a function of the quality of semantic information provided. Consequently, 11 of 12 participants were satisfied with their happy design, while only 2 of 12 participants were satisfied with their surprised design.

The participants also provided feedback about the limitations of our design system. Some participants found the automatic slider to be very aggressive since it made major changes in the motion, resulting in the loss of nuanced features of the motion. While parameter locking helps with capturing user intent about desired improvement and preserving nuances, it

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.

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

Figure 11: The evolution of the quality of user-designs (bars represent 95% CIs at each time step). The dotted slider represent the linear fit of mean scores over all emotions and participants, and the bands are a 95% CI around the fit.
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, a 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 owing to the sparse sampling of the dataset that powers the gallery.

**DISCUSSION AND FUTURE WORK**

Currently, our system allows design space exploration and editing given a single high-level semantic goal. Enabling concurrent design for multiple semantic design goals such as to express a mixture of emotions will provide the users with greater flexibility of design. Further, we will also explore if the user-design experience can be improved by capturing user-intent in more detail. Instead of using only high-level semantics to capture user-intent, high-level semantics could be further coupled with mid-level semantics relevant to the task. For instance, mid-level semantics corresponding to emotionally expressive motion design may correspond to actions such as dragging feet, jumping, or appearing crouched. Such a representation could also enable a more users to gain a better understanding of the design space.

Our system will also benefit from better dataset generation techniques. In particular, adaptive sampling which focuses on regions with better designs would allow the system to capture the design space with more fidelity. Additionally, on-demand sampling and dataset generation during user-design may further enable the system to provide guidance based on user-preferences. Re-using semantic information extracted from a particular robot’s dataset to enable the design for a different robot will also be essential for scalability of our system.

Finally, a simulation-driven design system like ours can only be as good as the underlying simulation. Our current motion parameterization and simulation doesn’t 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 and personality through motions only. However, aesthetics and physical structure are equally important for visual appeal. Parameterization and intuitive 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.

**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 succeeded at enabling desirable 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 catered to the task of allowing both novices and experts to create desirable robot motions.

**REFERENCES**


53. ROS.org | Powering the world’s robots. Retrieved April 4, 2018 from http://www.ros.org/

