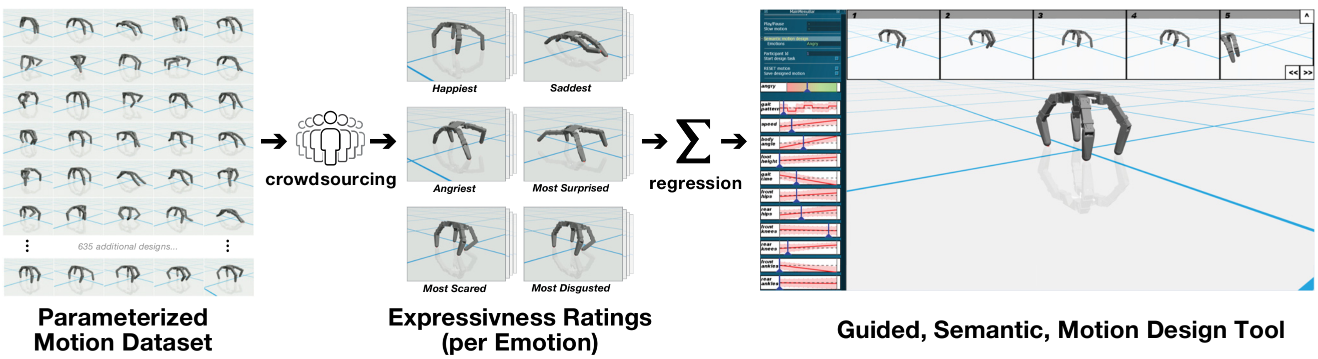
Geppetto: Enabling Semantic Design of Expressive Robot Behaviours

Ruta Desai[1], Fraser Anderson[2], Justin Matejka[2], Stelian Coros[1], James McCann[1], George Fitzmaurice[2], Tovi Grossman[2]

[1]Carnegie Mellon University, Pittsburgh, PA, USA.
{rutad, scoros, jmccann}@andrew.cmu.edu

[2]Autodesk Research, Toronto, ON, Canada.
{firstname.lastname}@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 crowdsourcing set-up 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
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

Submitted to UIST 2018

* *Figures 1, 2, and 7 of this document have been added as looping flash animations, viewable in Adobe Acrobat. If you experience any difficulties, please see our static version of this PDF, which has been submitted as supplementary material.*

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 anthropomorphic 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).

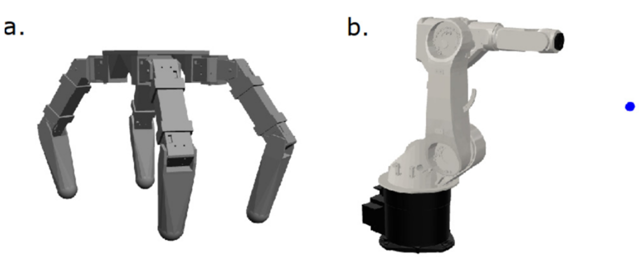


Figure 2: Users can design expressive motions for two distinct types of robots: (a) a quadruped, and (b) a robotic arm, while exploring the space of possible motions.

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 robotic systems for enhanced human-robot interaction [36, 42, 43]. Unfortunately, many of these principles are abstract and generic. They are therefore typically used either as add-on primitives for pre-existing motions [42], or as high-level guides for manual design, similar to how animators would use them [36]. The principles do not provide any guidance about synthesizing distinct emotive motions from scratch. Instead, our system enables users to design motions 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 [18, 37]. In particular, speed and robot pose [18, 20, 44], acceleration, and motion path curvature [4, 20, 37], and motion timing [20, 51] 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[1] instead of the joint space, inspired by how abstract trajectories could convey different emotions [4]. The system's semantically-guided parameter editing approach complements recent research on optimization-guided and keyframe-based motion editing for animated characters [22].

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.

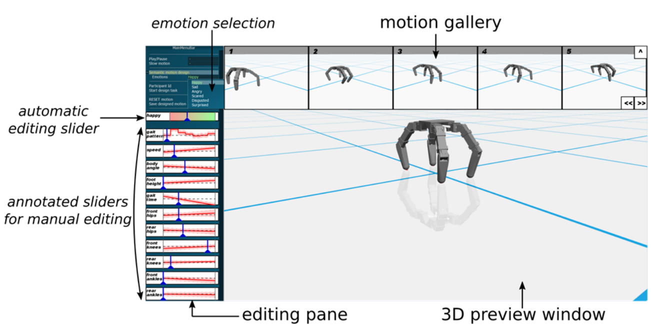


Figure 3: User interface overview. The 3D preview window renders the robot's motion. The gallery and annotated sliders provide semantically relevant information at design time.

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.

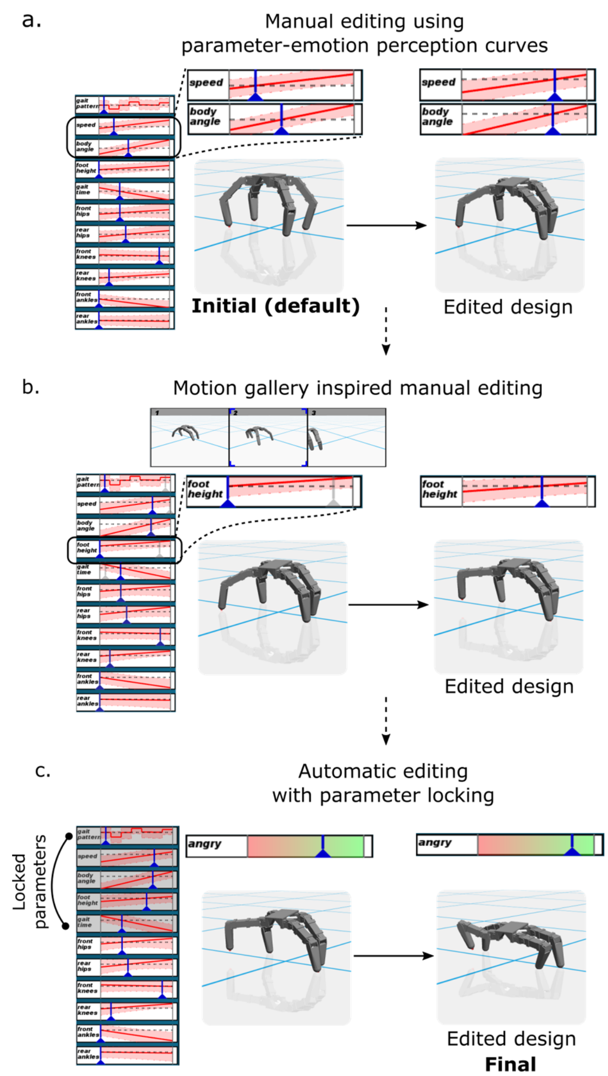


Figure 4: An example workflow designing an angry robot.

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.

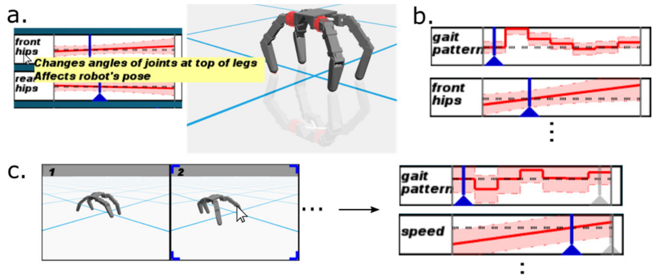


Figure 5: UI elements. (a) Parameter information is displayed as tooltips, and highlighted directly on the robot. (b) Parameter-emotion perception curve (in red) is visualized with an uncertainty band (shaded red) on each slider. The dotted line corresponds to current motion's estimated emotional perception. (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).

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.

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].

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.

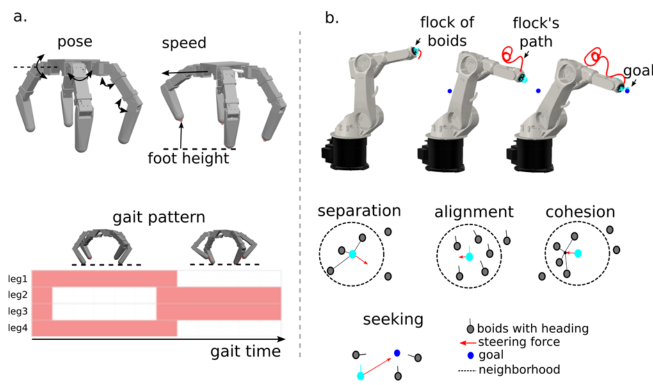


Figure 6: (a) The quadruped's motion is parameterized using joint poses, walking speed, foot height, gait time, and gait pattern (shown in red). (b) The arm is driven by a Boids flocking simulation. The flock is driven by interaction rules such as separation, cohesion, alignment, etc. The arm follows the resultant emergent path of the flock.

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.

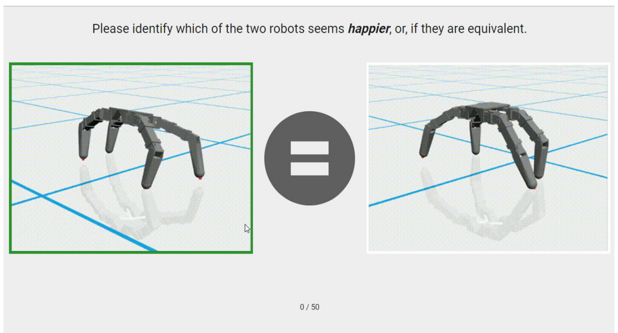


Figure 7: Interface crowd-workers used to judge emotion.

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 $N(\mu, \sigma)$ with mean μ, and uncertainty (σ). To compute the score, we create a modified Swiss-system style tournament [9] 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 [19] to convert the results of the tournament into Gaussian score estimates 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 one round of comparisons, wherein each sample is compared five times (against 5 different designs, by 5 different people), the designs ranked in the bottom half are eliminated. This process is repeated over three rounds (with five, five, and ten 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 or low-ranking samples. For the quadruped motion dataset there were a total of 3,355 comparisons to arrive at quality rankings for the top 25% of the samples. A more naïve 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.

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 7). Such a pairwise comparison approach has been preferred in the literature over asking the workers to provide an absolute score for individual samples [50].

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 φ_n, and a corresponding real-valued perception score μ, our goal is to learn a function $f : \varphi_n \rightarrow \mu$, 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 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 R^2 score of 0.50, and the worst-fitted (surprise) had $R^2 = 0.12$ (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 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 -- $\varphi_n = \{p_1, \ldots,$

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.

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 ($df/d\varphi$). The automatic update slider leverages this to update the robot motion. Unfortunately, since φ consists of both discrete and continuous parameters, we cannot compute the gradient $df/d\varphi$ 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 φ_c' is $\varphi_c' = \varphi_c + \delta \, df/d\varphi_c$, 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 [47]. δ is positive if the user moves the automatic slider to increase the emotional expression, and is negative otherwise.

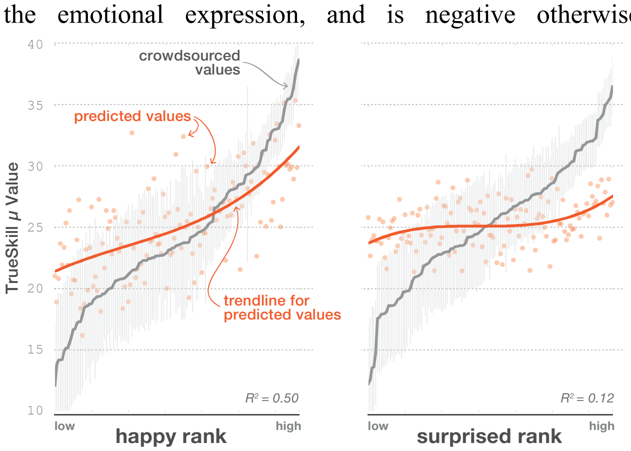


Figure 8: Comparison of predicted emotion values (orange) with their crowdsourced values (gray) for the test samples of the quadruped motion dataset. The best (happy) and worst (surprised) fitting emotion categories are displayed.

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.

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

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 to 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).

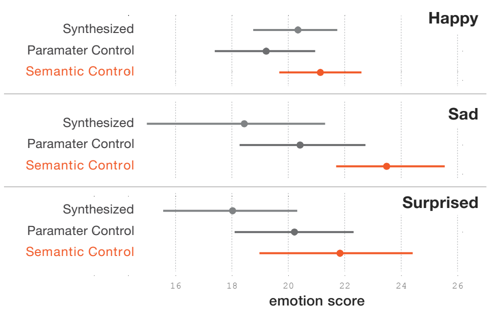


Figure 9: Mean emotion perception scores of the top 5 designs from the original dataset (*Synthesized*) with those created by the study participants. Bars show 95% CIs.

We also find that the designs created using the semantic UI outperform the best motions from our original dataset 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.

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

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).

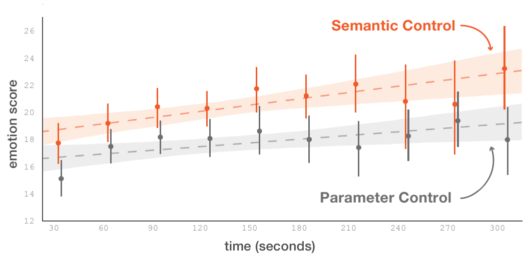


Figure 11: The evolution of the quality of user-designs (bars represent 95% CIs at each time step). The dotted lines represent the linear fit of mean scores over all emotions and participants, and the bands are a 95% CI around the fit.

Qualitative Feedback and Observations

The survey provided further insights about designing with 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.

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 dataset or individual predictor functions, participants used the automatic slider the most for this emotion.

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 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

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: e20989.
2. Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. 2016. Modeling stylized character expressions via deep learning. *Asian Conference on Computer Vision*, Springer, 136–153.
3. Connelly Barnes, David E. Jacobs, Jason Sanders, et al. 2008. Video Puppetry: A Performative Interface for Cutout Animation. *ACM SIGGRAPH Asia 2008 Papers* (SIGGRAPH Asia '08), ACM, 124:1–124:9. http://doi.org/10.1145/1457515.1409077
4. Lyn Bartram and Ai Nakatani. 2010. What makes motion meaningful? Affective properties of abstract motion. *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, IEEE, 468–474.
5. Cynthia Breazeal, Atsuo Takanishi, and Tetsunori Kobayashi. 2008. Social robots that interact with people. In *Springer handbook of robotics*. Springer, 1349–1369.
6. Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. 2013. Attribit: content creation with semantic attributes. *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM, 193–202.
7. Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 4302–4310.
8. Loïc Ciccone, Martin Guay, Maurizio Nitti, and Robert W. Sumner. 2017. Authoring motion cycles. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, 8.
9. László Csató. 2013. Ranking by pairwise comparisons for Swiss-system tournaments. *Central European Journal of Operations Research* 21, 4: 783–803.
10. 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. *Workshop on Autonomous Social Robots and Virtual Humans at the 25th Annual Conference on Computer Animation and Social Agents (CASA 2012)*.

11. Abhimanyu Dubey, Nikhil Naik, Devi Parikh, Ramesh Raskar, and César A. Hidalgo. 2016. Deep learning the city: Quantifying urban perception at a global scale. *European Conference on Computer Vision*, Springer, 196–212.
12. Magda Dubois, Josep-Arnau Claret, Luis Basañez, and Gentiane Venture. 2016. Influence of Emotional Motions in Human-Robot Interactions. *International Symposium on Experimental Robotics*, Springer, 799–808.
13. Victor Emeli. 2012. Robot learning through social media crowdsourcing. *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2332–2337.
14. Blender Foundation. blender.org - Home of the Blender project - Free and Open 3D Creation Software. *blender.org*. Retrieved April 4, 2018 from https://www.blender.org/
15. Vittorio Gallese, Christian Keysers, and Giacomo Rizzolatti. 2004. A unifying view of the basis of social cognition. *Trends in cognitive sciences* 8, 9: 396–403.
16. Madeline Gannon. 2017. *Human-Centered Interfaces for Autonomous Fabrication Machines*. Carnegie Mellon University.
17. 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: 144.
18. John Harris and Ehud Sharlin. 2011. Exploring the affect of abstract motion in social human-robot interaction. *RO-MAN, 2011 IEEE*, IEEE, 441–448.
19. Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. TrueSkill^{TM}: a Bayesian skill rating system. *Advances in neural information processing systems*, 569–576.
20. Heather Knight and Reid Simmons. 2016. Laban head-motions convey robot state: A call for robot body language. *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2881–2888.
21. Adriana Kovashka, Devi Parikh, and Kristen Grauman. 2012. Whittlesearch: Image search with relative attribute feedback. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2973–2980.
22. Yuki Koyama and Goto, Masataka. 2018. OptiMo: Optimization-Guided Motion Editing for Keyframe Character Animation.
23. Yuki Koyama, Daisuke Sakamoto, and Takeo Igarashi. 2014. Crowd-powered parameter analysis for visual design exploration. *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM, 65–74.
24. 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: 149.
25. Benjamin Lafreniere, Tovi Grossman, Fraser Anderson, et al. 2016. Crowdsourced fabrication. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM, 15–28.
26. Brian Lee, Savil Srivastava, Ranjitha Kumar, Ronen Brafman, and Scott R. Klemmer. 2010. Designing with interactive example galleries. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2257–2266.
27. 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. *International Workshop on Intelligent Virtual Agents*, Springer, 308–315.
28. Justin Matejka, Michael Glueck, Bradner, Erin, Hashemi, Ali, Grossman, Tovi, and Fitzmaurice, George. 2018. Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets. *SIGCHI (To appear)*.
29. 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: 216.
30. Brian K. Mok, Stephen Yang, David Sirkin, and Wendy Ju. 2014. Empathy: interactions with emotive robotic drawers. *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, ACM, 250–251.
31. Nikhil Naik, Jade Philipoom, Ramesh Raskar, and César Hidalgo. 2014. Streetscore-predicting the perceived safety of one million streetscapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 779–785.
32. Peter O'Donovan, Jānis Lībeks, Aseem Agarwala, and Aaron Hertzmann. 2014. Exploratory font selection using crowdsourced attributes. *ACM Transactions on Graphics (TOG)* 33, 4: 92.
33. Wei Pan and Lorenzo Torresani. 2009. Unsupervised hierarchical modeling of locomotion styles. *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 785–792.
34. Devi Parikh and Kristen Grauman. 2011. Relative attributes. *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 503–510.
35. Craig W. Reynolds. 1999. Steering behaviors for autonomous characters. *Game developers conference*, 763–782.
36. Tiago Ribeiro and Ana Paiva. 2012. The illusion of robotic life: principles and practices of animation for robots. *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, IEEE, 383–390.
37. Martin Saerbeck and Christoph Bartneck. 2010. Perception of affect elicited by robot motion. *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, IEEE, 53–60.
38. 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: 186.
39. Ari Shapiro, Yong Cao, and Petros Faloutsos. 2006. Style components. *Proceedings of Graphics Interface 2006*, Canadian Information Processing Society, 33–39.
40. Ronit Slyper, Guy Hoffman, and Ariel Shamir. 2015. Mirror puppeteering: Animating toy robots in front of a webcam. *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, ACM, 241–248.
41. Yuyin Sun. 2016. *Toward Never-Ending Object Learning for Robots*. PhD Thesis.
42. Daniel Szafir, Bilge Mutlu, and Terrence Fong. 2014. Communication of intent in assistive free flyers. *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, ACM, 358–365.
43. Leila Takayama, Doug Dooley, and Wendy Ju. 2011. Expressing thought: improving robot readability with animation principles. *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, IEEE, 69–76.
44. Haodan Tan, John Tiab, Selma Šabanović, and Kasper Hornb\a ek. 2016. Happy Moves, Sad Grooves: Using Theories of Biological Motion and Affect to Design Shape-Changing Interfaces. *Proceedings of the 2016 ACM*

Conference on Designing Interactive Systems, ACM, 1282–1293.
45. Frank Thomas, Ollie Johnston, and Walton Rawls. 1981. *Disney animation: The illusion of life*. Abbeville Press New York.
46. 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: 621–632.
47. Andreas Wächter and Lorenz T. Biegler. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106, 1: 25–57.
48. Jue Wang, Steven M. Drucker, Maneesh Agrawala, and Michael F. Cohen. 2006. The cartoon animation filter. *ACM Transactions on Graphics (TOG)* 25, 3: 1169–1173.
49. Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. A survey of crowdsourcing systems. *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.
50. 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: 86.
51. Allan Zhou, Dylan Hadfield-Menell, Anusha Nagabandi, and Anca D Dragan. 2017. Expressive robot motion timing. *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, ACM, 22–31.
52. Maya | Computer Animation & Modeling Software | Autodesk. Retrieved April 4, 2018 from https://www.autodesk.ca/en/products/maya/overview
53. ROS.org | Powering the world's robots. Retrieved April 4, 2018 from http://www.ros.org/
54. Today in Disney History: Lucky the Dinosaur Walks on the Scene | Disney Parks Blog. Retrieved April 1, 2018 from https://disneyparks.disney.go.com/blog/2013/08/today-in-disney-history-lucky-the-dinosaur-walks-on-the-scene/
55. aibo. *aibo*. Retrieved April 1, 2018 from http://aibo.com/
56. KUKA Robotics. Retrieved April 1, 2018 from https://www.kuka.com/en-ca