A Perceptual Control Space for Garment Simulation

Leonid Sigal[1] Moshe Mahler[1] Spencer Diaz[1] Kyna McIntosh[1] Elizabeth Carter[1]
Timothy Richards[2] Jessica Hodgins[1]
[1]Disney Research [2]Walt Disney Animation Studios

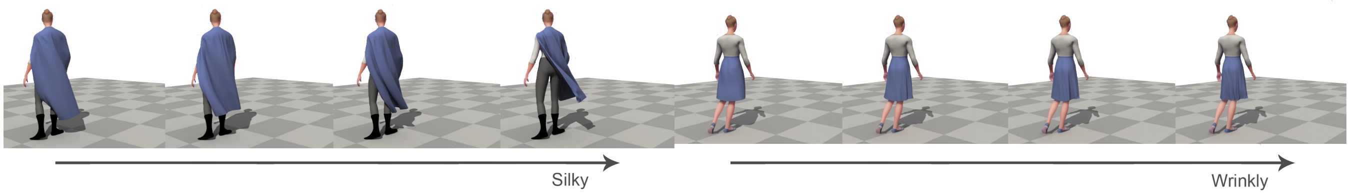


Figure 1: Perceptual Control Space: *The perceptual control model, learned from perceptual experiments conducted on a skirt, is illustrated on a skirt on the right and cape on the left. Left: Increasing levels of the* silky *trait. Right: Increasing the* wrinkly *trait.*

Abstract

We present a perceptual control space for simulation of cloth that works with any physical simulator, treating it as a black box. The perceptual control space provides intuitive, art-directable control over the simulation behavior based on a learned mapping from common descriptors for cloth (*e.g.*, flowiness, softness) to the parameters of the simulation. To learn the mapping, we perform a series of perceptual experiments in which the simulation parameters are varied and participants assess the values of the common terms of the cloth on a scale. A multi-dimensional sub-space regression is performed on the results to build a perceptual generative model over the simulator parameters. We evaluate the perceptual control space by demonstrating that the generative model does in fact create simulated clothing that is rated by participants as having the expected properties. We also show that this perceptual control space generalizes to garments and motions not in the original experiments.

CR Categories: I.3.7 [Three-Dimensional Graphics and Realism]: Animation— [I.2.10]: Vision and Scene Understanding—Perceptual reasoning

Keywords: perception, physical simulation, cloth simulation

1 Introduction

Cloth simulation can produce stunningly realistic examples of garments for virtual characters. The garments include detailed folding and wrinkling [Baraff and Witkin 1998; Bridson et al. 2003; Choi and Ko 2002; Grinspun et al. 2002], as well as other dynamic behaviors, for a variety of woven materials and knits [Kaldor et al. 2010]. The complex attire animated in recent feature films, such as *Brave* and *Frozen*, provides a concrete demonstration of the versatility of modern cloth simulators. However, art-directed control of the motion of the simulated clothing is still extremely difficult. Cloth simulators, such as Maya's nCloth and the proprietary programs used in feature production pipelines, contain tens or hundreds of parameters that modulate the dynamic behavior of the cloth and environment. These simulation parameters often loosely represent material properties and the various physical quantities of cloth, but there is no obvious mapping from the parameter labels to the common terms used to describe a physical piece of material or a garment, such as *softness*, *flowiness*, or *silkiness*.

While it is possible to measure physical parameters from samples of actual cloth [Miguel et al. 2012; Bhat et al. 2003], these parameters may not lead to the desired behavior because the parameters in the simulation model are an approximation of actual physical parameters. Moreover, dynamic parameters are particularly important to the quality of the resulting motion but are also the most difficult to measure. Further, fabric properties that do not necessarily correspond to a physical, measurable piece of fabric are often required to achieve a desired look for virtual clothing or scenery.

As a result, technical directors are tasked with the job of manually finding parameters that achieve a desired look by exploring the space of the simulator, often by sampling one or a few parameters at a time. Their job is further complicated because many parameters interact in complex ways to produce the final behavior of the garment. For example, parameters are not always independent, and there may be multiple ways to achieve the same look. Ideally, we would like to have controls for cloth simulators that are intuitive and perceptually meaningful, making the process of creating animated cloth faster and easier without reducing the ability to fine-tune the simulations as desired for a particular shot. We propose a methodology to achieve this goal by re-parameterizing cloth simulators to work with custom, perceptually discovered controls.

From conversations with technical directors, we learned that the terms used to critique their simulations were similar to the descriptions and adjectives used to describe cloth in the textile industry. For example, *china silk* is described by fabricdictionary.com as "lightweight and soft fabric", *chiffon* as "lightweight, extremely sheer, airy, and soft fabric", *taffeta* as "stiffened fabric with a crisp feel". The adjectives used in these descriptions, *lightweight*, *soft*, and *stiff*, can be interpreted as traits of the fabrics (using the nomenclature of [Matusik et al. 2002]). Such traits are intuitive to understand and can effectively span the space of both the real and the non-physical fabrics required for animation.

Equipped with this intuition, we build an easy-to-use and versatile approach for the control of cloth simulations using these traits as the axes of the control space. We perform a series of perceptual experiments and use the results in a linear sub-space regression analysis. The regression allows us to build a perceptual generative model over the simulation parameters, providing intuitive control over traits such as silkiness, flowiness, and lightness.

The perceptual experiments are performed in two phases. We first sample all of the potentially relevant parameters in the original simulator and run a series of user studies to measure the minimum step size along each of those parameters that results in a noticeable visual difference. This set of studies defines a perceptual unit for each parameter for the given simulator. We then use this unit to sample a second set of stimuli without bias. Another group of viewers are then tasked with ranking these stimuli on a Likert scale for one or more *traits* (*e.g.*, softness, snappiness, *etc.*). This second set of experiments allow us to learn a functional relationship between the traits as they are perceived and the original high-dimensional parameters of the simulator. We demonstrate the power of this approach by performing a third set of experiments to validate that incremental changes along the perceptual axes of these traits do indeed result in cloth appearances that match the selected levels of the traits. We further demonstrate how the traits can be combined, applied to other motions and garments, and used efficiently as an interface for tuning the look during the simulation process.

Contributions: First, we conduct a series of perceptual experiments to scale the parameters of a simulation to determine a perceptually valid "step-size" for each parameter. This set of experiments creates a common perceptual unit for each parameter independent of its native units (Nt/m for spring stiffness, for example). Second, we develop a method for re-parameterizing any cloth simulator in terms of a series of intuitive and meaningful traits and demonstrate the effectiveness of controlling cloth simulations in this space. While we learn the traits from a single motion (walking with a spin) and one sample garment (a skirt), we show that learned traits generalize and can be directly applied to other garments and motions. Although these perceptual experiments would need to be repeated for each new simulator, they do not require any understanding of the details of the simulation algorithm (other than a list of parameters) and the simulator can be treated as a black box.

2 Related Work

In this section, we review relevant work in the fields of computer graphics and vision. Our approach, which uses traits for control, is motivated by research that investigates perceptual controls for other problems and interfaces as well as by work examining attribute representations in computer vision. Additionally, we review cloth simulation and prior perceptual studies conducted on cloth dynamics.

Perceptual control and UI: Our approach is motivated by previous work that described a data-driven reflectance model [Matusik et al. 2002]. The authors acquired Bidirectional Reflectance Distribution Functions (BRDFs) from 100 real materials and then built a generative sub-space BRDF model, navigation through which was achieved via a set of perceptual attributes (that the authors refer to as *traits*, a term we borrow). Our goals and motivations here are conceptually similar; however, we work with cloth simulation parameters as opposed to BRDFs. This domain makes our perceptual setup more complex because we need to deal with a high-dimensional parameter space that has no, or at best a very complex, sub-space structure and requires dynamic video stimuli.

In a related effort, O'Donovan and colleagues [2014] proposed a perceptual trait-based user interface for exploration of a large collection of fonts. Similar to our study, MTurkers were tasked with comparing fonts along 37 traits. The resulting assessments were used to construct UI interfaces for *existing* font browsing; in contrast, our UI is a perceptual generative model. Expanding adjective mappings to crowd simulation, Guy and colleges [2011] established a linear mapping between personality descriptors from the Eysenck three-factor personality model and crowd simulation parameters.

Attribute-based representations: Attributes, that are human-nameable concepts, have recently become popular in computer vision. Their key benefit is that they allow for a human-centric, mid-level, shared representation in which categories can be efficiently represented using few attributal entities. Binary attributes have been used for image-based searches by object [Farhadi et al. 2009; Ferrari and Zisserman 2007], scene [Patterson and Hays 2012], face [Kumar et al. 2011] and sky descriptors [Tao et al. 2009]; more recent methods experimented with relative attributes [Kovashka et al. 2012; Parikh and Grauman 2011]. Laffont and colleagues [2014] introduced real-valued transient attributes that accounted for lighting, weather, season, and impression variations in outdoor scenes for use in image searching and editing.

Cloth simulation: Cloth simulation has a long history in computer graphics. Researchers have developed several different classes of approaches, including simple mass-spring systems [Baraff and Witkin 1998; Bridson et al. 2003], models that use continuum mechanics [Volino et al. 2009] and systems that model individual yarn structures [Kaldor et al. 2010]. These models have a variety of parameters, including material constants, that modulate their behavior. Spring-based systems typically have four parameters, including spring constants and damping coefficients, that modulate the response of the material to internal forces; continuum models have up to six parameters for planar deformation and bending. When these parameters are properly tuned, the resulting models can produce very realistic behaviors that closely match real fabrics [Miguel et al. 2012]. In this paper, our goal is not to improve these models in terms of quality of results or speed of simulation, but rather to address the issue of art-directed control, focusing on faster and intuitive tuning of parameters. In principle, any cloth simulator, even those yet to be developed, could be used as input for our approach.

Control of physical phenomena: There have been numerous works that address the controllability of physical phenomena, such as fluids, gases and cloth. In most cases, control relies on user-specified keyframes that serve as the targets for wave breaks [Mihalef et al. 2004], gases [McNamara et al. 2004], or clothing [Wojtan et al. 2006]. Also, a Bayesian optimization approach has been proposed to assist artists in parameter tunings through iterative procedural simulation and feedback [Brochu et al. 2010]. An interesting departure is the work of Cutler and colleagues [2007], where an art-directed system for the placement of wrinkles in cloth is developed. Such fine control is important for finessing individual shots and is complementary to the goals of this work, where we address the control of the overall dynamic behavior of the cloth as opposed to its appearance at a particular instant. Our approach can easily be combined with the aforementioned methods to provide both an overall look for the cloth and a match for specific keyframes.

Cloth parameter estimation: Prior work measured parameters for simulation directly from fabric samples. Some tried to extract parameters directly from video [Bhat et al. 2003; Bouman et al. 2013; Wang et al. 2011]. Bhat and colleagues [2003] used optimization to estimate simulation parameters that minimized discrepancies between simulated and observed fabric. Bouman and colleagues [2013] modified this approach by using a data-driven regression model based on the marginal statistics of image features computed from video. While these approaches are compelling due to their simple and inexpensive acquisition, they may have difficulty separating the internal material-specific parameters from the external environmental parameters [Miguel et al. 2012]. To this end, custom-built apparatuses have been proposed to account for external forces on the fabric and allow estimation of parameters for a variety of models [Miguel et al. 2012; Wang et al. 2011]. While such methods are powerful and can allow the creation of a library of fabric presets for a given simulation, their weakness is that they can only measure the behavior of real fabrics. Therefore, they may be inappropriate for scenarios where virtual characters move in unrealistic ways or a desired look does not match existing textiles.

Perceptual studies of cloth: Bouman and colleagues [2013] performed the first and, to our knowledge, only perceptual study of humans' ability to estimate the material properties of fabric from images and video. In their experiments, they showed participants two real fabrics side by side, under the same and different amounts of wind, and asked which fabric was stiffer or denser on a 7-point scale. Similar experiments were done from static image stimuli. They found that people's perception from video correlated well with the ground truth stiffness and density measured by a commercial textile research laboratory, but perception from images did not, resulting in a hypothesis that people tend to use dynamic cues for the task. We expand upon this work by having viewers assess numerous traits of animated fabrics from video.

McDonnell and colleagues [2006] performed a set of perceptual experiments to validate the use of level-of-detail (LOD) models for cloth simulation. They evaluated the efficacy of impostors that used cloth simulations at various resolutions in crowd simulations. In a similar approach to ours, they performed perceptual experiments to capture people's judgments of material properties (in their case, stiffness). We largely confirm their findings by showing that changes in stiffness are more perceivable in softer materials.

3 Method Overview

Our goal is to learn a perceptually meaningful control space, for an arbitrary simulator, to provide intuitive, art-directable control. To this end, we show participants stimuli comprised of rendered instances of the real simulator parameter settings and ask them to rank the stimuli for a particular perceptual trait of interest (*e.g.*, *flowiness*). From this data, a mapping from the average rating of *flowiness* and the original simulator parameter set can be learned, resulting in the ability to specify *flowiness* as a new perceptual one-dimensional control. This process can be replicated for other traits.

The task is complicated, however, by the fact that the original parameter space is high-dimensional and highly non-linear. This property implies that uniform sampling of the original parameter space is likely to produce biases in the learned mapping. For example, if people are insensitive to a parameter, small steps may result in imperceptible differences and result in a mapping that learns that the parameter is unimportant. However, it may be very important for bigger steps because of large-range or non-linear effects.

To alleviate these issues, we divide the problem into two parts with corresponding perceptual experiments. First, we learn a perceptible unit of measure along each dimension in the simulator parameter space to create a perceptual metric. We also can use this information to observe to which simulator parameters participants are more sensitive, providing a lower bound of perceptibility for a given parameter change in the simulation pipeline. Second, we use this metric to sample stimuli and have participants rank them along one or more perceptual dimensions on a Likert scale. We use these rankings to learn a mapping from the perceptual control space to that of the original simulator.

Parameter	Dependence
Density	×
Stretch Stiffness	×
Stretch Stiffness Max	Stretch Stiffness
Stretch Damping	×
Stretch Percentage Min	×
Stretch Percentage Max	Stretch Percentage Min
Shear Stiffness	×
Shear Stiffness Max	Shear Stiffness
Shear Damping	×
Shear Angle Min	×
Shear Angle Max	Shear Angle Max
Bend Polar Stiffness	×
Bend Polar Stiffness Max	Bend Polar Stiffness
Bend Damping	×
Bend Polar Radius 1	×
Bend Polar Radius 2	Bend Polar Radius 1
Dynamic Damping	×
Air Density	×
Air Drag Inside	×
Air Drag Outside	Air Drag Inside
Lift Inside	Lift Outside
Lift Outside	×

Parameter
Bend Angle Dropoff
Bend Resistance
Bounce
Damp
Deform Resistance
Drag
Lift
Stickiness
Stretch Resistance
Tangential Drag
Thickness

Figure 2: Simulator Parameters: *In our experiments, we make use of two high-quality cloth simulators: a custom production pipeline simulator [Fabric 2013] and Maya's nCloth simulator. In both cases, we use a subset of parameters selected through consultation with professional simulation artists. In the case of [Fabric 2013], we additionally tie some of the parameters to one variable, as noted in the table. Parameters noted with the × symbol are free, while those that are tied are designated by the parameter to which they are tied; in most cases, the tied parameter is twice the parameter value to which it is tied.*

3.1 Experiment: Perceptual Units

The goal of the perceptual metric learning is to encode how similar or dissimilar a change in a simulator parameter may appear to an average observer. In principle, the approach to this is simple: sample stimuli in the space of simulator parameters, show pairs of stimuli to participants, and ask them how similar they perceive the stimuli to be. The difficulty arises from two key challenges: (1) the simulator parameter space is very high-dimensional and (2) the metric is likely not global (*e.g.*, [R. McDonnell et al. 2006] show that stiffness is more perceivable in soft materials). To address these two challenges, we make two assumptions. First, we assume that we can approximate the perceptual metric space locally by examining a sparse set of neighborhoods around a set of hand-designed presets (*e.g.*, cotton, wool, leather). Second, we assume an axis-aligned metric space that aligns with the original simulator parameters. Together, these two assumptions allow us to learn the perceptual metric space with a tractable number of perceptual experiments.

3.1.1 Participants

We recruited adult research participants using Amazon Mechanical Turk (MTurk)[1], with a minimum of ten participants per task. This research was approved by our Institutional Review Board, and participants were compensated for their time. In order to participate in this research, participants had to be 18 years of age or older and have self-reported normal or corrected-to-normal vision.

3.1.2 Stimulus Generation

First, we create a varied set of P *presets* for the simulator (in our experiments, $P = 5$ for [Fabric 2013]). These presets define points in the high-dimensional simulator parameter space around which local metrics will be learned. To create the presets, we asked professional animators to create parameter settings for the simulator that result in the appearance of a varied set of fabrics: cotton, wool,

[1] For a pilot study, we ran similar studies in both a controlled laboratory setting and on MTurk. We found that the performance was similar, with the exception of MTurk results typically containing a few outliers.

silk, leather, and chiffon. The fabrics are then animated on a knee-length skirt worn by a character performing a runway walk cycle.

We use 22 parameters from a custom high-quality production pipeline simulator [Fabric 2013], given in Figure 2 (left). These parameters were selected through consultation with professional technical simulation artists and included a subset of the parameters in the software. Based on consultation and documentation, we also coupled some parameters together, resulting in a set of 14 independent parameters that we sample. Tying two parameters together implies that one was always expressed as a fraction or multiple of another. While here we write all of the explanations of stimuli in terms of a custom simulator [Fabric 2013], we have also run similar experiments with Maya's nCloth [Stam 2009], using the parameters in Figure 2 (right), to demonstrate the versatility of our experimental design. For each parameter p, we sample N_p times at increments between the minimum and maximum values provided in the documentation. Depending on the range of available values for a specific parameter, we take either exponential or linear deterministic steps in our sampling. We apply these parameter values to each of the five preset fabrics for our experiments.

3.1.3 Procedure

We created a total of 580 videos (5 presets $\times$ 116 videos each). For each parameter, we had between 4 and 17 movies (with most around 7). For each preset/parameter combination, we created an online experiment using a web survey service, SurveyGizmo, to which participants were directed. The order of trials was randomized for each participant, and participants provided informed consent and verified that they could see the videos before beginning.

Each trial included two videos, presented one above the other, with the question, "Do the skirts in the two videos appear to move the same or differently?" and multiple choice checkboxes for "Same" and "Different". Participants saw each of the 4 to 17 movies compared to the original movie six times; the original movie was also shown compared to itself 12 times. The vertical order of the videos was counterbalanced, and the order of trials was randomized for each participant. To ensure that the participants focused on the overall percept of the fabric as opposed to minute differences, such as the exact folding pattern, we ensured that they could not pause the video or see a static frame at any time. We also allowed them to watch the videos at most two times, although the participant could answer the question any time before the two repetitions completed.

3.1.4 Results and Discussion

Each experiment had a minimum of ten participants with successful completion; additional participants were excluded if they failed to complete the experiment or they had outlying data, reporting that they saw a difference in a large (95%) fraction of trials. This pattern suggests that a respondent was not paying attention because there were 12 trials where the two movies were identical. For the data from the remaining participants, we performed two-sided paired t-test analyses to compare responses to the questions with two identical videos to those with two different videos (*i.e.*, the preset versus a specific sample value). We consider the deviation significant for p values lower than 0.05, henceforth referred to as a perceptually visible significant difference (PVSD) in the parameter setting.

Figure 3 illustrates the compilation of results, where green bars indicate the parameter value for each preset and red bars indicate PVSDs at $p = 0.05$ significance. We note three behaviors of interest. First, changes in some simulator parameters (*e.g.*, Shear Angle or Shear Damping) are imperceptible (at least for the skirt and the walk motion that we used). Second, for some parameters (*e.g.*, Density or Bend Polar Stiffness Min/Max), the PVSD moves with the

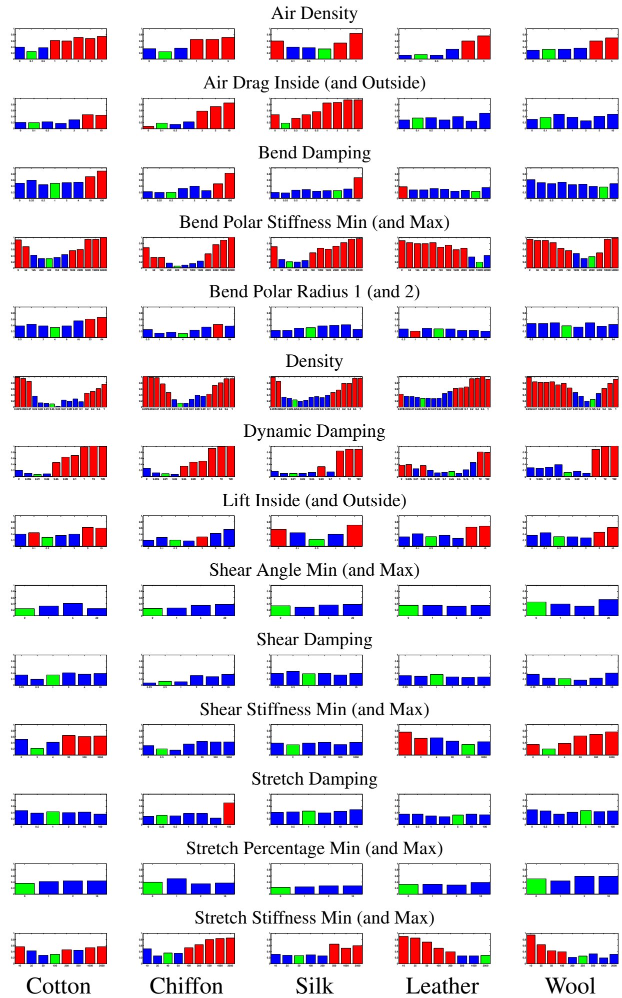
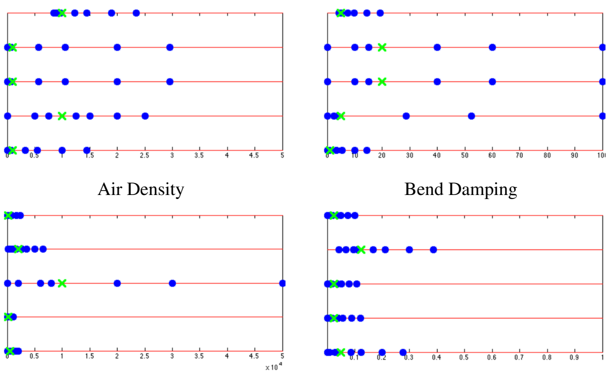


Figure 3: Perceptibility of the simulator parameters: *Our results for the five presets and fourteen simulator parameters tested. Bar heights indicate the fraction of participants that observed a difference between the value on the x-axis and the corresponding preset value (denoted by the green bar). Red bars indicate parameter settings that illustrate a significant, perceivable effect ($p < 0.05$).*

preset value by scaling linearly (Density) or exponentially (Bend Polar Stiffness Min/Max). This observation implies that a relatively simple functional metric for this particular parameter may be sufficient across the entire parameter space. The exponential scaling observed for Bend Polar Stiffness supports previous findings [R. McDonnell et al. 2006]. Third, in some cases, a simulator parameter change becomes imperceptible only when other parameters are set such that the fabric lies in a different part of the simulator space. One example is Air Drag, which is imperceptible for heavier fabrics (*e.g.*, leather or wool) but extremely perceptible for lighter fabrics (*e.g.*, silk). Interestingly, Shear Stiffness Min/Max seems to exhibit the opposite behavior, where it becomes imperceptible for lighter fabrics and more perceptible for heavier ones.

We believe that the experiments in Figure 3 themselves are informative, at the very least as an informal guide to which parameters simulation artists should target when creating fabrics for characters. However, our core goal is to define a perceptual metric space over simulation parameters. Given the experiment described above, the actual metric is then defined by finding the PVSD threshold on both the positive, δ_i^+, and negative, δ_i^-, sides of the change required in the parameter i that results in the first PVSD.



Air Density Bend Damping

Bend Polar Stiffness Min (and Max) Density

Figure 4: Nine-sided die sampling with PVSD: *Each red line indicates a possible range (x-axis) of the parameter for a given material preset (y-axis). The presets, from top to bottom, in each plot are chiffon, wool, leather, silk, and cotton. The value of the parameter for each of the five material presets is marked by a green* x*, and the* $\{-1.5, -1.0, -0.5, -0.25, 0.25, 0.5, 1, 1.5\}$ *multipliers of PVSD are illustrated by blue points. Jointly, these plots illustrate the marginal sampling for four (of fourteen) parameters from which stimuli for perceptual trait ranking experiment are generated.*

3.2 Experiment: Perceptual trait ranking

After defining the PVSD thresholds, $\{\delta_i^-, \delta_i^+\}$, for each parameter, we create a second experiment to learn the interactions between various parameter settings and perceptual effects in semantic terms. This axis learning is achieved by sampling regions around the five presets and allowing MTurk users to rank them along selected traits.

3.2.1 Participants

With similar requirements as in the previous experiment, we recruited a minimum of 20 participants per task using MTurk.

3.2.2 Stimulus Generation

We start with each of the five material presets created for the first experiment and vary the settings for each of the parameters by rolling an unloaded, 9-sided die with sides corresponding to multiples of $\{-1.5, -1.0, -0.5, -0.25, 0, 0.25, 0.5, 1, 1.5\}$. The result tells us the fraction of the PVSD threshold to add or subtract from the preset settings for each parameter. Note that rolling the die independently for each parameter has a cumulative effect, so the expected deviation of the sampled stimuli from the original preset is close to $(|-1.5|+|-1.0|+|-0.5|+|-0.25|+0+0.25+0.5+1+1.5)/9 \times 14$ parameters, which is approximately 10 PVSD. We illustrate the effect of rolling a die and sampling in Figure 4. There is more variability in the samples around the presets than there is among the presets themselves. The material presets tend to differ in many simulator parameters; the effect is cumulative, allowing the desired look to be achieved with relatively small deltas, along each parameter, from the presets. When estimating the PVSD, we vary one parameter at a time, so a larger step is needed to perceive a difference.

We created 135 videos by rolling the die for each parameter 26 times and including the original preset. We removed the videos that failed to simulate or produced a bad-looking simulation result (*e.g.*, high frequency jitter due to instability), leaving 81 videos.

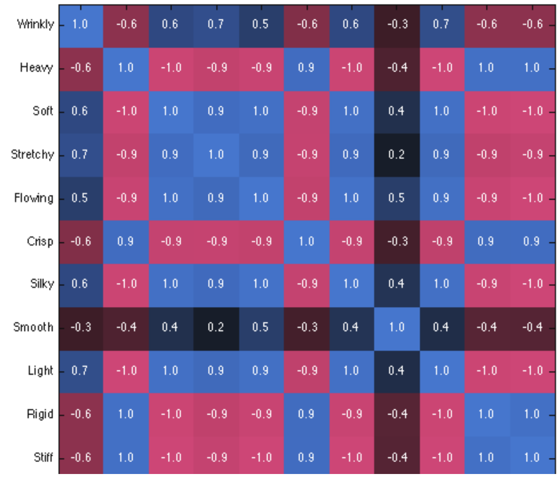


Figure 5: Correlations among traits: *Blue corresponds to positive and red to negative correlations among viewer responses; brighter colors correspond to stronger correlations/inverse correlations with actual correlation coefficients inscribed in each square. Note that many traits appear to be semantic synonyms or antonyms.*

3.2.3 Procedure

We created a survey for each of eleven common traits: *wrinkly*, *heavy*, *soft*. *stretchy*, *flowing*, *crisp*, *silky*, *smooth*, *light*, *rigid*, and *stiff*. To obtain these traits, we used a fabric dictionary[2] to gather a set of common terms that describe the dynamic behavior (rather than the appearance) of fabrics. We also augmented this list with traits suggested by several technical directors. All 81 videos were included in each survey, and participants viewed each and answered the question "How <trait> is the material of the skirt?" on a 5-point Likert scale that ranged from "1 = Very <trait>" to "5 = Not <trait>". The order of trials was randomized for each participant, and participants provided informed consent and verified that they could see the videos before beginning a survey.

3.2.4 Results and Discussion

First, we averaged the scores for each video and trait combination across all participants. We sorted the videos based on their average score rankings to determine whether the scores corresponded to reasonable levels of each trait, as shown in Figure 6.

Our visualizations resulted in two observations. First, the average participant ranking for videos does appear to correspond appropriately to specific traits. Second, the mapping from a specific trait to the parameter settings is often complex and multi-modal. For example, the positive value of a trait, X, can be more informative than the negative value of the trait, not X. Specifically, something that is *not black* is not necessarily *white*; it could be any of a variety of colors. Similarly, something that is *not soft* could possess any of a variety of other traits. Furthermore, there may be variability even in the exemplars for the positive value of a single trait. We found that this was true of the *wrinkly* trait, as shown in Figure 6. Two of our stimuli were rated as very wrinkly (value of 4.5 / 5.0). One of these stimuli was a lightweight, silk-like fabric that wrinkled while still being airy and flowing. The other was more similar to a heavy cotton and tended to look clingy rather than flowing.

Additionally, we were able to analyze the semantic similarity of our trait set because the same videos were used to assess all eleven

[2] `http://www.fabricdictionary.com`

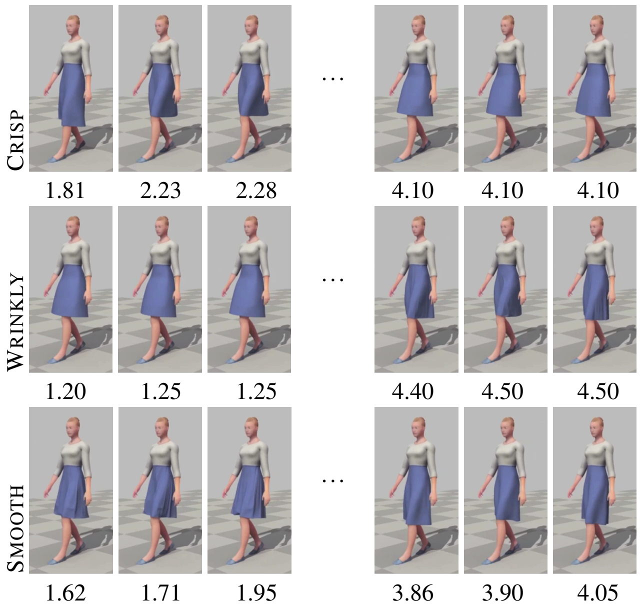


Figure 6: Examples of trait ratings: *Ratings produced by participants in perceptual trait ranking experiment. Frames from the stimuli that received the lowest and highest three ratings (on a Likert scale of 1 to 5) are presented with their average ratings.*

traits. The correlations between each pair of traits are shown in Figure 5; positive correlations are in blue, negative correlations are in red, and exact correlation coefficients are inscribed in each cell of the matrix. Some of our traits are highly positively correlated, suggesting semantic similarity, whereas others are negatively correlated, opposing each other. The results align with real-world experience: certain traits, such as *silky* and *flowing*, are often used together to describe the same piece of fabric; others, such as *stiff* and *soft*, are not. We further analyzed these results by performing unsupervised clustering with correlations treated as affinities. We use Affinity Propagation [Frey and Dueck 2007] to discover groups of highly correlated traits ($p = 0.9$), including

{ wrinkly },
{ soft, stretchy, flowing, silky, light },
{ heavy, crisp, rigid, stiff }, and
{ smooth }.

With a stricter parameter setting ($p = 0.95$), we can separate { stretchy } and { crisp } into their own clusters, suggesting that they have subtly different semantic interpretations from the parameters with which they are grouped at $p = 0.9$.

The most important outcome of this analysis is that we can expect the learned perceptual control axis for the simulator to be highly non-orthogonal and, in some cases, highly correlated. This result is not surprising, as similar behavior was observed in other perceptual parameterizations. For example, Matusik and colleagues [2002] found that BRDF traits are also non-orthogonal. We note that while the perceptual axes that we will learn are inherently non-orthogonal, that does not make them any less effective for control, although it does suggest a UI design where the user is able to see the correlations so that he or she is not tempted to produce meaningless settings (*e.g.*, fabric that is *light* but not *silky*).

We want to highlight that the above results should not be interpreted as suggesting that the perceptual space of cloth is four-dimensional. There could be other traits that we have not tested that may span the space in other ways. Further, the specific motion of the character in our experiment may render certain dynamic properties and traits unobservable. The simulator itself is also limited in generating only certain fabrics, not all of which correspond to or adequately represent real textiles. The observation that the control space is effectively four-dimensional is interesting and was not something we initially anticipated, but does not imply that we found a universal and complete perceptual space for the description of the dynamic behavior of fabric. Rather, what we found is that many commonly used trait names are redundant in a perceptual sense. Some animators may choose to use different trait names for the same behaviors, or define other custom traits that internalize their mental picture of cloth behavior and may span the perceptual space in other ways.

3.3 Learning simulator re-parameterization

Given the set of simulator parameter settings $\mathbf{y}_i \in \mathbb{R}^{14}$ (for stimuli i from the trait ranking experiment) and corresponding perceptual ratings $\mathbf{x}_i \in \mathbb{R}^{11}$ for the 11 traits used[3], we want to learn a functional mapping between the values of any subset of the trait parameters and the 14-dimensional simulator parameters. Because the traits are generally non-orthogonal, we also must incorporate into our UI a method to ensure that user is aware of the effect that control over one trait has on other traits. This UI awareness ensures that a user does not attempt to control the simulation in an inconsistent way (*e.g.*, asking for both *light* and *heavy*). These properties motivate the use of sub-space linear regression for the functional mapping because other forms of regression do not typically deal gracefully with missing values in the input vector $\mathbf{x}$. We first learn a joint low-dimensional manifold across paired data $\{\mathbf{x}_i, \mathbf{y}_i\}$, $i \in [1, ..., 81]$ and then show how we can use it to control individual perceptual traits or any trait combination. We use Singular Value Decomposition (SVD) to pull out the principal directions of variations, $\mathbf{U}$, allowing us to represent the joint vector by

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{U}\mathbf{v} + \mathbf{b}, \tag{1}$$

where $\mathbf{v} \in \mathbb{R}^d$ are coefficients where for all experiments in this paper, we used $d = 10$, which accounts for 90% of the variance in the data. Note, $d = 15$ (with full space being $d = 25$) accounts for 99.7% of variance, due to the correlations among the elements in $\mathbf{x}_i$ as noted in the analysis of perceptual trait ranking experiment.

Given this model and a user-supplied value for *all* of the traits $\mathbf{x}$, we solve for subspace coefficients with a least-squares pseudo-inverse:

$$\mathbf{v}^* = (\mathbf{U}^T P^T P \mathbf{U})^{-1} \mathbf{U} P^T (\mathbf{x} - P\mathbf{b}) \tag{2}$$

where

$$P = \begin{bmatrix} \mathbf{I}_{11\times 11} & \mathbf{0}_{11\times 14} \\ \mathbf{0}_{14\times 11} & \mathbf{0}_{14\times 14} \end{bmatrix} \tag{3}$$

is a selection matrix. The corresponding simulator parameters $\mathbf{y}^*$ can then be obtained by:

$$\begin{bmatrix} \mathbf{x}^* \\ \mathbf{y}^* \end{bmatrix} = \mathbf{U}\mathbf{v}^* + \mathbf{b}. \tag{4}$$

It is unreasonable to assume, however, that the user would want to supply all values for different perceptual traits, so this model can be generalized for any subset of the parameters by simply re-defining P to be a sparser version with a non-zero element only at the diagonal entry(ies) corresponding to user-specified attribute/trait value(s). For example, to enable control for the 1st and 3rd traits,

$$P = \begin{bmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \mathbf{0}_{3\times 22} \\ \mathbf{0}_{22\times 3} & \mathbf{0}_{22\times 22} \end{bmatrix}, \tag{5}$$

[3] Note that for convenience, we normalize the perceptual ratings, such that each element of $\mathbf{x}_i$ is a real-valued number between 0 and 1.

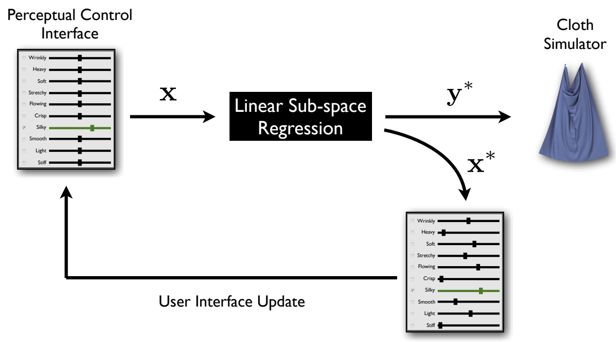


Figure 7: Illustration of Perceptual Control: *User adjustment to the trait* silky *is shown on the left (in green). User control is encoded in vector* $\mathbf{x}$ *and propagated through a learned linear subspace regression model to produce estimates corresponding to simulator parameters,* $\mathbf{y}^*$*, and the feedback for the UI,* $\mathbf{x}^*$*. As* silky *is increased, correlated traits (*e.g.*,* flowing *and* soft*) also increase.*

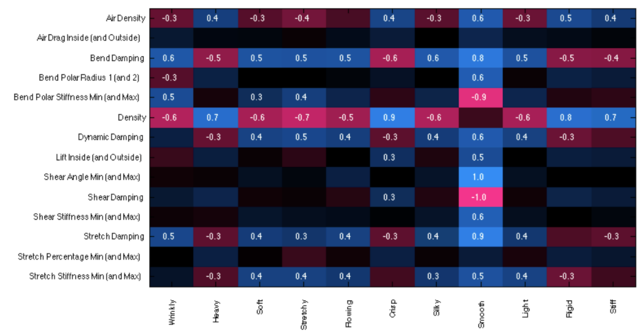


Figure 8: Relationship among learned perceptual traits and original simulator parameterization: *Normalized derivatives (-1 to 1) of the original simulator parameters (y-axis) with respect to perceptual traits (x-axis) for the learned model. Blue indicates a positive derivative (e.g., as* heavy *increases, so does* density*) and negative derivatives are red. The relative value of the derivative whose magnitude* ≥ 0.25 *is in the corresponding square; we do so to highlight the strongest effects. The presence of a number indicates a strong relationship between the learned perceptual trait and the corresponding simulator parameter.*

and $\mathbf{x} = [\alpha_1, 0, \alpha_3, \mathbf{0}_{1\times 8}]^T$ would specify the corresponding trait values $0 \leq \alpha_i \leq 1$ chosen by the user. Note that, in that case, the subspace solution will also give us values for the unspecified parameters for the various other traits (in $\mathbf{x}^*$), making it easy to see the effect in the perceptual simulator control space for a more intuitive UI. The illustration of the process is given in Figure 7. As a final post-processing step, we clamp the predicted simulator parameter values, $\mathbf{y}^*$, to a valid range to ensure that all values are within acceptable bounds for the simulator.

It is interesting to probe the learned model to see what correlations exist between the perceptual parameter traits and the original parameters of the simulator. In Figure 8, we plot a visualization of the derivatives of the original parameters of the simulator with respect to each perceptual trait. Because the relationship between the two parameterizations is linear, the derivative is a constant and represents the correlation between the perceptual parameter and original simulator parameterization. To enhance visualization, we normalize the derivative to be between -1 and 1 by dividing all derivatives by the maximum derivative magnitude. In Figure 8, we see strong correlations between perceptual parameters and various sets of physical parameters. For example, perceptual *light* inversely correlates with physical *density* and directly correlates with *bend damping*, but does not correlate strongly with *lift* or *shear*.

We have also experimented with other forms of regression, including nearest neighbor, kernel and linear regression. We found nearest neighbor regression to be highly unstable, in terms of control, with behaviors flipping from one mode to another. Kernel regression was more stable, but with a large enough kernel resembled a simple linear regression model. Linear regression resulted in largely similar performance to our method in most instances. However, these alternative regression methods do not directly provide a way to estimate the values of unspecified perceptual parameters, which we believe makes them less appealing to use in practice. They also do not provide a consistent means of moving between the original simulator parameter space and the perceptual parameter space, which allows for a much more streamlined, homogeneous pipeline.

Post-processing: Some changes in the traits and parameters of the original simulator cause the garment to stretch longer than the designed length. This artifact is a common problem with any simulator (*i.e.*, relaxing the stiffness, necessary for light or stretchy fabrics, results in stretching of virtual material due to gravity). These effects make side-by-side visual comparisons of different fabrics and trait values less effective. To address this problem, we measure the length of the fabric at rest, using the geodesic distance from the top of the garment to the bottom, and then correct for deviations from the designed length by adjusting the rest length (geometry) of the garment in the gravity direction. In other words, if we see that parameters make the fabric 10% longer, we make the geometry 10% shorter in the gravity direction. In our experience, this procedure is very effective and allows us to achieve fixed-length garments irrespective of parameter settings. Rest length adjustments do not alter the trait appearance in the resulting cloth simulations.

4 Evaluation

We validate the learned perceptual control space, as well as the paradigm for creating it, in a number of perceptual experiments that test the quality of the learned model, the ease of use, and the integration with existing workflows. First, viewers rated whether videos created using our trait axes properly corresponded to those traits. Second, we show how the learned control space can be customized for individual users and used seamlessly in a collaborative workflow. Third, we compared the ease of use of our perceptual control interface to the traditional simulator control interface.

4.1 Assessing the quality of the learned perceptual simulator parameterization

We generate the stimuli by selecting five or six values along each trait axis and rendering a total of 59 videos. As in our earlier experiments, we create surveys in which participants are shown one video at a time, asked "How <trait> is the material of the skirt?", and given a 5-point Likert scale from "1 = <Not trait>" to "5 = <Very trait>". Videos were presented in a random order, and participants viewed all 59. Twenty individuals from MTurk successfully participated in this study after providing informed consent and confirming that they could view the videos.

We illustrate results in Figure 9. For each trait, the top row shows the sampled values as a percentage of the trait (from 0 to 100) along the perceptual control axis, and the bottom row shows the corresponding average viewer ratings. While we formulate the control in Section 3.3 in terms of fraction of the trait present (between 0

							Correlation
CRISP	30	37	44	50	60	70	0.94
	1.6	1.8	3.3	3.6	4.3	4.4	
FLOWING	30	50	60	70	80	90	0.97
	1.2	1.9	2.4	3.8	4.3	4.7	
HEAVY	20	30	40	50	60	80	0.94
	1.8	2.6	3.3	3.7	4.2	4.3	
LIGHT	30	50	60	70	80	85	0.96
	1.5	2.0	2.2	3.2	3.9	4.2	
SILKY	50	60	70	80	86	90	0.96
	2.4	2.3	3.6	4.2	4.4	4.5	
SMOOTH	30	60	65	70	75	80	0.97
	2.0	3.8	3.7	3.8	4.3	4.2	
SOFT	30	50	60	80	85	90	0.96
	1.7	2.0	2.5	4.4	4.7	4.9	
STIFF	20	40	50	60	80	90	0.89
	1.3	3.7	4.1	4.5	4.9	4.9	
STRETCHY	30	60	70	80	86	90	0.99
	1.2	2.7	3.5	4.3	4.7	4.4	
WRINKLY	30	60	80	90	100		0.94
	1.0	2.1	4.7	4.6	4.5		
						Average	**0.95**

Figure 9: Evaluating perceptually learned trait axes: *For each trait, sample values used to create stimuli are shown in the top row and the corresponding participant ratings from a 5-point Likert scale are presented in the bottom row. We sample values that best exemplify behavior along a trait. To measure the perceptual quality of our model, we calculate correlations, averaging 0.95.*

and 1), for easier exposition, we talk about it here it in terms of percentage of trait value present by multiplying trait controls by 100. We calculated the correlations between the two values; high correlations indicate that our model successfully captured the percept of the trait for the fabric. For all traits, we achieve high correlations (mean = 0.95) and participants usually were able to rank the stimuli successfully along the trait axes. One exception was for *stretchy*, for which participants thought that 86% *stretchy* was stretchier than 90% *stretchy* as defined by our axis. However, we note that the two stimuli are only 4% apart and are difficult to visually distinguish.

4.1.1 Generalization of learned perceptual control to other garments and motions

Although we developed our trait control space using perceptual experiments with a single motion and garment type, the control space is general and is useful for controlling fabric behavior for other garments and motion types. To illustrate this effect, we show control traits applied to a different garments, a cape (Figure 1) and suspended cloth (Figure 10). The cape is naturally less constrained than the skirt, so the overall motion appears looser. However, the desired perceptual effect is observable as the parameters progress through each trait axis. For example, the cape looks progressively silkier as the value of *silky* increases incrementally in Figure 1 from 50% to 80%. These examples are also shown in the video, as well as the effect of the trait applied to a different motion (running). Again, the trait properties are observable.

4.1.2 Effect of mesh resolution and time stepping

While the mesh resolution of the garment and the global parameters of the simulator (*e.g.*, time step) typically have a profound effect on the overall simulation of the fabric, these effects are largely independent of the learned perceptual space because they globally affect the garment motion and do not directly influence the control directions. The alternative would mean that the original controller parameters would similarly change with mesh resolution and timestamp (*e.g.*, Air Drag would no longer act as Air Drag when the

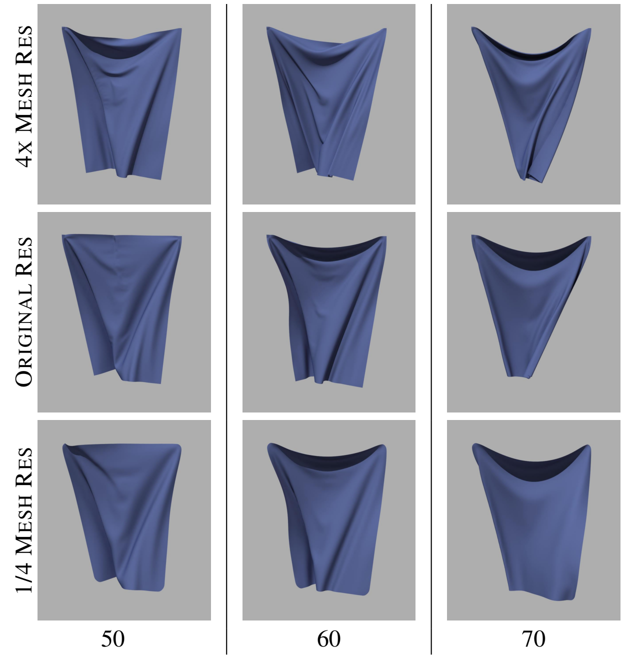


Figure 10: Effect of changing resolution (flowing trait on suspended cloth): *One frame from each simulation is illustrated; resolution increasing bottom to top, flowing trait increasing left to right.*

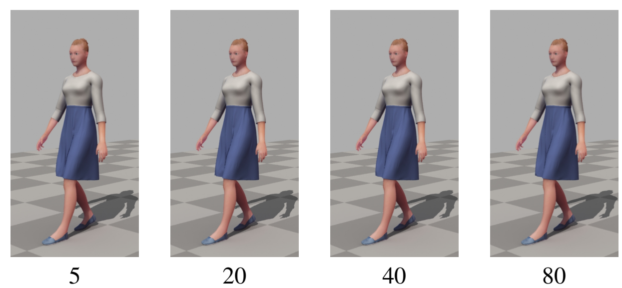


Figure 11: Effect of changing time-step (wrinkly trait, 80%): *The time-step (number of collision sub-frames) ranges from 5 to 80; for all other experiments, we always use a value of 20.*

resolution of the mesh is altered), which is not the case in most simulators, including ours.

However, we illustrate these effects because they change the global offsets for both our perceptual control space and the original simulator control space. The results for resolution are illustrated in Figure 10 where we up-sample and down-sample the original mesh of the suspended cloth (note the model was learned with a skirt). Our original mesh resolution for the cloth is 8,192 triangles; for the $4\times$ mesh resolution, we up-sample the mesh to 32,768 triangles, and for 1/4 mesh resolution, we down-sample to 2,048 triangles. Overall, it is clear that wrinkles are more visible at higher resolutions; however, the *flowing* trait remains a viable control across all resolutions. Our simulator is very stable so the time-step (in our simulator, the *number of collision sub-frames*) has little effect on the quality of the simulation (Figure 11). The supplemental video includes footage of both tests and additional examples.

4.2 Combining perceptual controls

One powerful feature of our model is the ability to seamlessly combine any subset of the traits to achieve a desired look. We include examples in the supplemental video.

4.3 Creating custom perceptual control axes

Another benefit of our approach is that anyone can easily create custom perceptual controls by ranking stimulus movies on a multi-point scale. A similar idea was explored by Nikolaus Troje for human kinematic animations [Troje 2015]. Every animator on a production can have his or her own set of perceptual controls while retaining the ability to share the created content. This functionality is enabled by the fact that learned perceptual controls map back to the original simulator parameter space and vice versa.

To illustrate that we can adequately learn controls from a single participant (as opposed to a group of MTurk participants), we had a single animator perform perceptual trait ranking for two traits, *heavy* and *soft*. We observed that the perceptual controls created by the animator were largely similar to the ones we were able to learn from the sample average, despite having much less data.

4.4 Collaborative workflow integration

One of the benefits of our model is that it can seamlessly work within current simulation pipelines by integrating with existing assets or simulator presets that were created using the original parameterization. As we discussed, we have an explicit bidirectional mapping between the original simulator parameters and the perceptual controls. Consider animators A and B collaboratively working on related shots, each having his/her own custom perceptual controls created using the proposed methodology. Animator A can create the look he/she likes and save it by converting it to the original simulator parameter space; Animator B can then convert the file into his/her own perceptual parameterization to make alterations.

Note, in the *collaborative* regime, we store everything in the original simulator parameter space and apply correctives in the perceptual spaces. This strategy ensures that dimensions orthogonal to any one perceptual space remain at their specified values and are not *lost* in the edits; however, we also provide an option, in the interface, to override them if a user desires. Both of these behaviors are useful, depending on the intention of the collaborators.

To illustrate this collaborative functionality, we take a *cotton* preset created using the original controls and make it lighter by projecting it into the perceptual control space and dialing up or down the *light* control by 10%. The result is presented in the supplemental video.

4.5 Ease and efficiency of control

In this work, our assumption is that a perception-based interface for simulation control would be more intuitive and easier to use than commonly available interfaces. To validate this assumption, we ran a small-scale user study with two novice Maya animators. We asked each animator to create skirts to match two reference skirts using both the original interface [Fabric 2013] and our perceptual interface. We counterbalanced the order of the interface used. Neither animator had experience with either interface, and both interfaces use the same underlying simulator engine [Fabric 2013]. The reference fabrics $\mathcal{A}$ and $\mathcal{B}$ were created in the same simulator so that it was possible to recreate them exactly. Because we assume that pre-visualization is the major time cost in garment parameter tuning, we recorded the number of pre-visualizations required for the simulation results prior to settling on a solution. The results are illustrated in the supplemental video and analyzed in Figure 12.

To determine how comparable the artists' results were using the two systems, we performed a survey. Thirty participants were recruited using the same qualifications and method used previously. For the stimuli, we created ten video pairs: two where each reference video was compared to itself and eight where an artist's result was compared with the appropriate reference. Each pair was shown twice in random order. Viewers were instructed to look at the skirts and rank the similarity of the materials in the two videos using Likert scales (1-identical, 5-extremely different). The mean ratings and standard deviations for similarity were 1.50(0.50) for the identical pairs, 2.40(0.67) for the traditional UI pairs, and 2.38(0.64) for the perceptual UI pairs. An ANOVA indicated a significant main effect of creation type (F = 39.43, p <0.0001), though there was no significant pairwise comparison for perceptual versus traditional methods (p >0.05). Therefore, our perceptual interface performed equivalently for visual quality. In terms of efficiency, however, our interface is faster, despite comparable numbers of degrees of freedom (14 for the original interface after tying parameters, and 11 for the perceptual interface). The animators required less than half as many pre-visualization runs in the perceptual interface.

	Animator 1		**Animator 2**		
	Reference		Reference		
	$\mathcal{A}$ 	$\mathcal{B}$ 	$\mathcal{A}$ 	$\mathcal{B}$ 	Total
Traditional	14	8	9	4	35
Perceptual	4	2	6	4	16

Figure 12: Efficiency of control: *We asked two novice animators to match fabric simulations illustrated in $\mathcal{A}$ and $\mathcal{B}$ using both simulator interfaces. The table illustrates the number of pre-visualization runs they had to perform in the process. They achieved similar results in both, but ours required fewer pre-visualizations.*

5 Conclusion and Discussion

In this paper, we present a perceptual control space for physical simulation of cloth that works in conjunction with any physical simulator. Our perceptual control space provides intuitive, art-directable control over the simulation based on a learned mapping from common terms (traits) for describing cloth to control parameters of a given simulator. We learn this perceptual control space from a series of perceptual experiments assessing the traits of the simulated fabric. Our methodology provides a number of benefits. First, as we show, the learned control space directly corresponds to percepts of resulting simulated fabric and generalizes to other motion and garments. This mapping allows for easy and intuitive control over simulation so that the animators in our user study achieve desired results faster. Second, it works within existing simulation pipelines and the developed methodology is applicable to any simulator. Through our experimental design, we were also able to gain insight into which parameters produce noticeable changes in the simulations, which we hope will inform animators.

We want to highlight that while the raw dimensionalities of the two parameterizations (perceptual and original) are similar, the dimensions have very different meanings and control efficacy. To achieve a desired appearance in the original simulator parameter space, the user would change many of the 14 parameters. However, a given fabric in the proposed perceptual trait parameterization can be obtained using one or two of the 11 traits. For example, one easily can create a *light* or *light* and *flowing* fabric, but achieving a similar look in the original simulator parameterization would require specifying many parameters. Our re-parameterization allows easy control and is the core advantage of our method. We find that one or two traits are often sufficient to describe dynamic behaviors.

While we show that our controls generalize to other motions and garments, we have not explored what effects those factors may have on perception. For practical reasons, we limited our stimuli to mid-length skirt on a woman performing a walk cycle with a turn. More widely varying stimuli may potentially exercise richer dynamic behaviors of the simulator. Further, while we assume a linear relationship between the traits and underlying simulator parameters, which worked very well, a potentially better mapping can be learned using non-linear sub-space regression models (*e.g.*, Shared KIE [Memisevic et al. 2012]). In similar vein, it may be useful to explore perceptual units in other spaces. In particular, based on preliminary observations, log space may be useful for certain parameters.

We expect that this methodology will be predominantly used to customize controls and that it will significantly reduce the amount of time it takes to create a garment that exhibits a desired trait. We provide evidence that customization would allow animators to more efficiently create cloth simulations with specific traits, and we intend to perform further experiments to address this issue on a larger scale and examine the use of such controls in production workflow.

References

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. *ACM SIGGRAPH*.

BHAT, K. S., TWIGG, C. D., HODGINS, J. K., KHOSLA, P. K., POPOVIC, Z., AND SEITZ, S. M. 2003. Estimating cloth simulation parameters from video. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

BOUMAN, K. L., XIAO, B., BATTAGLIA, P., AND FREEMAN, W. T. 2013. Estimating the material properties of fabric from video. In *International Conference on Computer Vision*.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

BROCHU, E., BROCHU, T., AND DE FREITAS, N. 2010. A bayesian interactive optimization approach to procedural animation design. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

CHOI, K., AND KO, H. 2002. Stable but responsive cloth. *ACM Transactions on Graphics 21*, 3, 604–611.

CUTLER, L. D., GERSHBEIN, R., WANG, X. C., CURTIS, C., MAIGRET, E., PRASSO, L., AND FARSON, P. 2007. An art-directed wrinkle system for CG characters clothing and skin. *Graphical Models 69*.

FABRIC. 2013. Walt disney animation studios production simulator.

FARHADI, A., ENDRES, I., HOIEM, D., AND FORSYTH, D. 2009. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

FERRARI, V., AND ZISSERMAN, A. 2007. Learning visual attributes. In *Neural Information and Processing Systems (NIPS)*.

FREY, B. J., AND DUECK, D. 2007. Clustering by passing messages between data points. *Science 315* (February), 972–976.

GRINSPUN, E., KRYSL, P., AND SCHRODER, P. 2002. Charms: A simple framework for adaptive simulation. *ACM Transactions on Graphics 21*, 3, 281–290.

GUY, S. J., KIM, S., LIN, M. C., AND MANOCHA, D. 2011. Simulating heterogeneous crowd behaviors using personality trait theory. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2010. Efficient yarn-based cloth with adaptive contact linearization. *ACM Transactions on Graphics 29*, 4.

KOVASHKA, A., PARIKH, D., AND GRAUMAN, K. 2012. Whittlesearch: Image search with relative attribute feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*.

KUMAR, N., BERG, A., BELHUMEUR, P., AND NAYAR, S. 2011. Describable visual attributes for face verification and image search. *IEEE TPAMI 33*, 10.

LAFFONT, P.-Y., REN, Z., TAO, X., QIAN, C., AND HAYS, J. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics 33*, 4.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2002. A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3.

MCNAMARA, A., TREUILLE, A., POPOVIC, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Transactions on Graphics 23*, 3, 449–456.

MEMISEVIC, R., SIGAL, L., AND FLEET, D. J. 2012. Shared kernel information embedding for discriminative inference. *IEEE TPAMI 34*, 4.

MIGUEL, E., BRADLEY, D., THOMASZEWSKI, B., BICKEL, B., MATUSIK, W., OTADUY, M., AND MARSCHNER, S. 2012. Data-driven estimation of cloth simulation models. *Computer Graphics Forum 31*, 2.

MIHALEF, V., METAXAS, D., AND SUSSMAN, M. 2004. Animation and control of breaking waves. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

O'DONOVAN, P., LIBEKS, J., AGARWALA, A., AND A. H. 2014. Exploratory font selection using crowdsourced attributes. *ACM Transactions on Graphics 33*, 4.

PARIKH, D., AND GRAUMAN, K. 2011. Relative attributes. In *International Conference on Computer Vision*.

PATTERSON, G., AND HAYS, J. 2012. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*.

R. MCDONNELL, S. DOBBYN, S. C., C. O'SULLIVAN R. MCDONNELL, S. DOBBYN, S. C., MCDONNELL, C. O. R., DOBBYN, S., COLLINS, S., AND O'SULLIVAN, C. 2006. Perceptual evaluation of LOD clothing for virtual humans. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

STAM, J. 2009. Nucleus: Towards a unified dynamics solver for computer graphics. *IEEE International Conference on Computer-Aided Design and Computer Graphics*, 1–11.

TAO, L., YUAN, L., AND SUN, J. 2009. Skyfinder: Attribute-based sky image search. *ACM Transactions on Graphics 28*, 3.

TROJE, N., 2015. www.biomotionlab.ca/demos/bmlwalker.html.

VOLINO, P., MAGNENAT-THALMANN, N., AND FAURE, F. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics 26*, 3.

WANG, H., O'BRIEN, J. F., AND RAMAMOORTH, R. 2011. Data-driven elastic models for cloth: Modeling and measurement. *ACM Transactions on Graphics 30*, 4.

WOJTAN, C., MUCHA, P. J., AND TURK, G. 2006. Keyframe control of complex particle systems using the adjoint method. *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.