# Active Learning for Interactive Audio-Animatronic® Performance Design

Joel Castellon and Moritz Bächer
Disney Research Los Angeles

Matt McCrory, Alfredo Ayala, and Jeremy Stolarz
Walt Disney Imagineering

Kenny Mitchell
Disney Research Los Angeles and Edinburgh Napier University

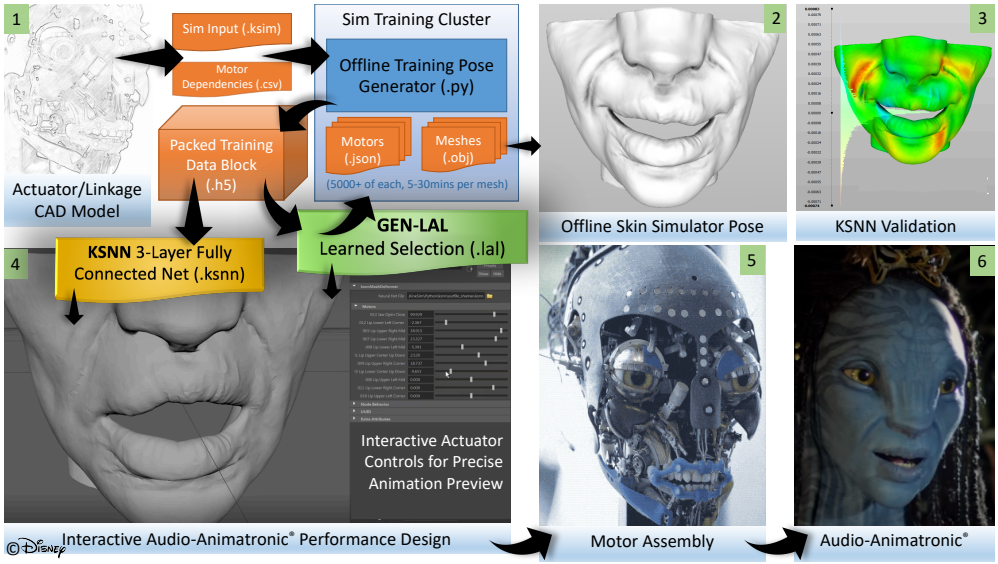



**Figure 1**. Successful Audio-Animatronic® figures often require many costly physical fabrication and assembly revisions.Our virtual production workflow, however, takes the CAD model data (1) and computes hyperelastic material simulations driven from motor activations to digitally preview Audio-Animatronic® pose meshes (2). With these meshes, we train our *KSNN* solver to reconstruct in real-time with accuracy to within 834µm (3). Our Learning Active Learning *GEN-LAL* scheme adaptively learns to select pose inputs for accelerated training convergence. With training costs halved and interactive animation design (4), the final assembly (5) can be completed more efficiently with a higher quality of experience for guests (6).

## Abstract

We present a practical neural computational approach for interactive design of Audio-Animatronic® facial performances. An offline quasi-static reference simulation, driven by a coupled mechanical assembly, accurately predicts hyperelastic skin deformations. To achieve interactive digital pose design, we train a shallow, fully connected neural network (KSNN) on input motor activations to solve the simulated mesh vertex positions. Our fully automatic synthetic training algorithm enables a first-of-its-kind learning active learning framework (GENLAL) for generative modeling of facial pose simulations. With adaptive selection, we significantly reduce training time to within half that of the unmodified training approach for each new Audio-Animatronic® figure.

## 1. Introduction

Imagineers Lee Adams, Roger Broggie, Leota Toombs, and Wathel Rogers are credited with developing the first Audio-Animatronics® toward Walt Disney's vision to inspire and delight guests by making inanimate things move on cue and come to life, hour after hour and show after show. From the first humanoid figure "Little Man" in 1951 [Gluck 2013], and the "A1" birds developed for the *Enchanted Tiki Room* in 1962, to the recent "A1000" figures found in *Star Wars: Galaxy's Edge* [Blitz 2019], Nishio et al.'s "Geminoids"™[2010], and Garner Holt Productions' expressive "Lincoln" [Holt 2017], the goal of delivering autonomous physical characters out of the uncanny valley [Mori 1970] is a core creative and scientific endeavor increasingly necessitating accurate interactive performance-design tools.

The design of today's Audio-Animatronic® figures [Blitz 2019], using trial and error on mutually dependent physical motor configurations and coupled skin material fabrication iterations, can lead to large costs or yield unsatisfactory uncanny robotic performances. This work addresses the facial performance design task by simulating the physical skin material motions resulting from rotational motor activations and yields an interactive design workflow with high accuracy to the actual *real world* Audio-Animatronic.®

We form a first-of-a-kind production system for interactive facial performance design of stylized, hyper-realistic, and humanoid figures. Our contributions across the major virtual prototyping system components of a simulator, KSNN and GENLAL, are as follows:

- *Coupled assembly-skin solver*: An accurate offline simulator for expressive skin material shapes uniquely coupled with facial motor actuators is developed. Our novel quasi-static, coupled solver, accurately predicting states of assembly and flexible skin components is introduced in Section 4.

- *Neural assembly-skin solver*: A shallow, fully connected neural network, KSNN, synthetically trained with the offline simulator for interactive pose-prediction en-

abling our interactive design workflow is proposed in Section 5. As a somewhat less challenging process, our solver is potentially also applicable to high-quality digital-only facial character animation, for example, in video games or movie virtual production.

- *Adaptive training*: A new generative Learning Active Learning framework, GENLAL, for skin-pose predictions with equivalent speed and accuracy to KSNN using only half the number of training samples is detailed in Section 6.

We follow with related work in Section 2. An overview of the interactive Audio-Animatronic® performance design system is covered in Section 3 before detailing each of the system subtopics. The article ends with limitations of the methods (Section 7), concluding remarks, and future work (Section 8).

## 2. Related Work

*Fabrication-oriented design.* There is a body of work on the design and fabrication of mechanical assemblies [Zhu et al. 2012; Ceylan et al. 2013; Coros et al. 2013; Thomaszewski et al. 2014; Bächer et al. 2015], compliant structures [Megaro et al. 2017], deformable objects made of rubber-like [Skouras et al. 2013] or silicone materials [Zehnder et al. 2017], or robotic characters [Megaro et al. 2015; Bern et al. 2017; Geilinger et al. 2018]. Closest to our work is the physical cloning process proposed by Bickel et al. [2012]. However, unlike these works, our focus is on the accurate simulation and animation of an *existing* Audio-Animatronic's® head. Rather than estimating the actuator motions for more than 100 target facial-expression poses offline, our method, rather, interactively predicts the simulated physical facial-mesh shapes from motor-assembly states for performance pre-visualization. In our simulations, we couple the silicone skin to a mechanical assembly, modeling the traditional actuation of a Audio-Animatronic's® facial expressions. While Bickel et al. [2012] focus on accurate cloned surface shapes and do not provide performance statistics of their forward-simulation generating-target surfaces offline, our neural solver operates in real-time (Section 6.3).

*Facial-performance simulation.* A related physical facial-performance simulation work can be found in *Phace* [Ichim et al. 2017], which simulates the physical muscle and skin-tissue shape of a human face under internal and external forces, but is not purposed for Audio-Animatronic® actuated skin material and is not real-time. Barrielle and Stoiber [2019] compute a real-time performance-driven inertial and *sticky lips* simulation, but deal with motion input from facial landmarks of an actor rather than rotational motors of an Audio-Animatronic.® Instead of accelerating physical face simulation, Bailey et al. [2018] provide a data-driven process for fast approximation of rigged virtual-character animations by combining linear skinning and learned

local non-linear deformations. Their non-linear per-bone deformation neural networks are trained with a random selection of poses within a local range of motion of each bone, whereas our performance-design method trains a base-pose-relative facial network by adaptively generating new training poses through *Active Learning*, reducing training times by half.

*Active Learning.* Active Learning (AL) encompasses situations where we are particularly concerned about the selection of batches of training instances sequentially.[1] The motivation behind this setup is to either accelerate training convergence or to actively compensate for class imbalance. Successful applications include experimental drug design [Warmuth et al. 2003] (where getting an example amounts to carrying out a complex chemical experiment), object recognition [Sivaraman and Trivedi 2014], and text classification [Tong and Koller 2002].

However, most research in AL focuses on classification problems. Thus, the most well-known heuristics for variable selection (uncertainty sampling, expected largest model change, query by committee; see [Settles 2009] for more details) require probability estimates over a discrete set of classes. Nonetheless, there are AL algorithms that deal with variable selection with continuous output [Krause et al. 2008] that derive in the well-known uncertainty sampling heuristics (e.g., aim for uncertainty and novelty). Krause et al. [2008], however, make strong assumptions that might be specific for sensor temperature data (e.g., Gaussian process modeling), and the resulting algorithm is directly derived from those (e.g., entropy functions are submodular).

There is also work that tries to reduce AL problems to outlier-detection type of algorithms by deriving an upper bound on the generalization error [Sener and Savarese 2018]. While outlier-detection algorithms are efficient and simple to implement, they also rely on some mathematical assumptions of the model generating the predictions. For example, Sener and Savarese [2018] proves the bound for CNNs function characteristics applied to classification. In contrast, our work is the first of it's kind to apply active-learning concepts to the accelerated training of activation-based animation-solving in the area of generative modeling.

## 3. Audio-Animatronic® Performance Design

Traditional approaches to Audio-Animatronic® design and longevity testing are time-consuming and expensive. Rod puppets provide engineers with real-world testbeds, but these require the design, fabrication, and assembly of a puppet's mechanical elements as well as the formulation, modeling, and pouring of a silicone skin. Once a puppet has been built, a review of the design can be conducted, the outcome of which informs changes to the next iteration.

[1] This is sometimes referred to as batch mode Active Learning.

In the simulator (see Section 4), we developed a physically accurate software tool capable of recreating this workflow in the digital realm, greatly reducing iteration time and cost. Entire head mechanisms can be designed and assembled, and different formulations of skin can be quickly tooled and modeled, all in a virtual space. Coupling finite-element (FE) degrees of freedom to the states of assembly components, we are then able to compute how a silicone skin will stretch, bend, and fold across the assembly, based on the positioning of virtual actuators.

Due to the high degree of accuracy in these solves, an individual mesh-pose simulation on a modern multi-threaded CPU can take between five and 30 minutes of compute time to converge. While this represents a dramatic speed-up from the traditional physical Audio-Animatronic® design approach, downstream stages of the production pipeline, including animation, require real-time interactivity.

Learned models offer the potential to augment this workflow with trained neural networks (see Figure 1), capable of delivering accurate results without sacrificing interactivity (see Section 5).

The workflow begins in computer-aided design software (Figure 1.1). The design is iterated over using the simulator to import the mechanisms and simulate the resulting deformations of the silicone skin. Upon completion of the design, a physical Audio-Animatronic® head is assembled (Figure 1.5), and the custom silicone skin is then formed and mounted to the head (Figure 1.6). Meanwhile, the simulator computes a set of simulated poses offline (Figure 1.2) which, in turn, feed our learned prediction models for fast approximation (Figure 1.3; see supplementary material and Figure 6 for detailed error comparisons). With the learned KSNN model, we are able to perform interactive pose editing with an artist-friendly digital content-creation-package plugin, feeding into the fully realized Audio-Animatronic® (Figure 1.6, see the accompanying supplementary video).

## 4. Audio-Animatronic® Skin Simulator

To deform synthetic skin into an expressive set of facial expressions, the skin is commonly attached to a mechanism driven by a set of rotational motors. To make adaptive learning a tractable endeavor, we devised a simulator that can accurately predict skin deformations under a particular configuration of motors.

For assembly simulation, we rely on a constrained optimization-based formulation [Coros et al. 2013]:

$$\mathbf{c}(\mathbf{s}) = 0,$$

where the state vector $\mathbf{s} \in \mathbb{R}^{6m}$ represents the rotational and translational degrees of freedom (DoFs) of the $m$ assembly components. To formulate constraints that restrict the relative motion between pairs of components, we define frames that remain constant in local component coordinates. We then transform them to global coordinates

by applying the rigid transformations encoded in the state vector, formulating a set of constraints between centers and pairs of axes. For example, for a hinge that connects two components, the two frame centers have to coincide in global coordinates, and the transformed $z$-axis of the frame of the first component has to remain orthogonal to the transformed $y$- and $z$-axes of the frame of the second component, assuming the hinge axis to equal the $z$-axis. Among joints of varying DoFs, motors take on a special role. We can think of them as hinges, where the relative angle between components is prescribed. In simulations, we therefore step rotational motors, then solve the set of non-linear equations for the component states that fulfill all joint constraints.

Since Audio-Animatronic® skin is typically made of silicone or a related material, we rely on a hyper-elastic material for its simulation. The skin is rigidly attached to skull-like shells in some regions. Moreover, to translate assembly motion into skin deformations, the skin is attached to the assembly at a discrete set of locations. For instance, the mouth-corner deformation is driven by several attachment points to the assembly.

To prepare for simulations, we first generate a conformal, volumetric mesh. For the deformed configuration, we differentiate between nodes that are rigidly moving with assembly components, and hence are expressed with a rigid-body transformation extracted from the state vector $\mathbf{s}$, and nodes whose deformation is due to the elastic response of the skin. To solve the assembly state and the skin nodes $\mathbf{x} \in \mathbb{R}^{3n}$ that can freely move, we formulate and solve a quasi-static, constrained problem,

$$\min_{\mathbf{s},\mathbf{x}} \int_V \Psi(\mathbf{s},\mathbf{x})\,\mathrm{d}V \quad \text{s.t. } \mathbf{c}(\mathbf{s}) = 0,$$

where $\Psi$ is the strain-energy density of the hyper-elastic material [Sifakis and Barbic 2012]. Note that, in this coupled formulation, the assembly can have passive DoFs that are attached to and regularized by the deforming skin.

Holding partial degrees of freedom fixed with further fixed displacements, the skin-strain energy is modeled in discrete elements, which we integrate according to the density of the hyperelastic material over the volume. We have experimented with an increasing level of complexity when it comes to the order of elements and model complexity of the hyper-elastic material. The use of linear elements is clearly insufficient to achieve the desired precision. For simulations of the silicone material, we observe best performance when using a Mooney-Rivlin or Yeoh material model [Sifakis and Barbic 2012]. For minimization, we use sequential-quadratic programming (SQP) [Nocedal and Wright 2006]. Because the skin is attached to rigid shells, and at a sufficiently dense set of locations to the underlying mechanism, quasi-static simulations are sufficient.

## 5. KSNN: Learned Interactive Mesh Solving

With a potentially large number of motor actuators (ranging from nine to 40 in facial motor assemblies) and non-linear physically dense mesh configurations, we found the application of simple fitting schemes were lacking in their physical accuracy and required corrective elements to bring them back into an acceptable range. Further, linear regression or principle component analysis (PCA) were also considerably insufficient. Our goal to be computationally fast with high accuracy led us to veer away from ensemble regression schemes [Fanelli et al. 2013; Kazemi and Sullivan 2014], as those perform with lower accuracy for our particular domain. Given the input motor actuators forming an unstructured collection of independent linear rotations (except

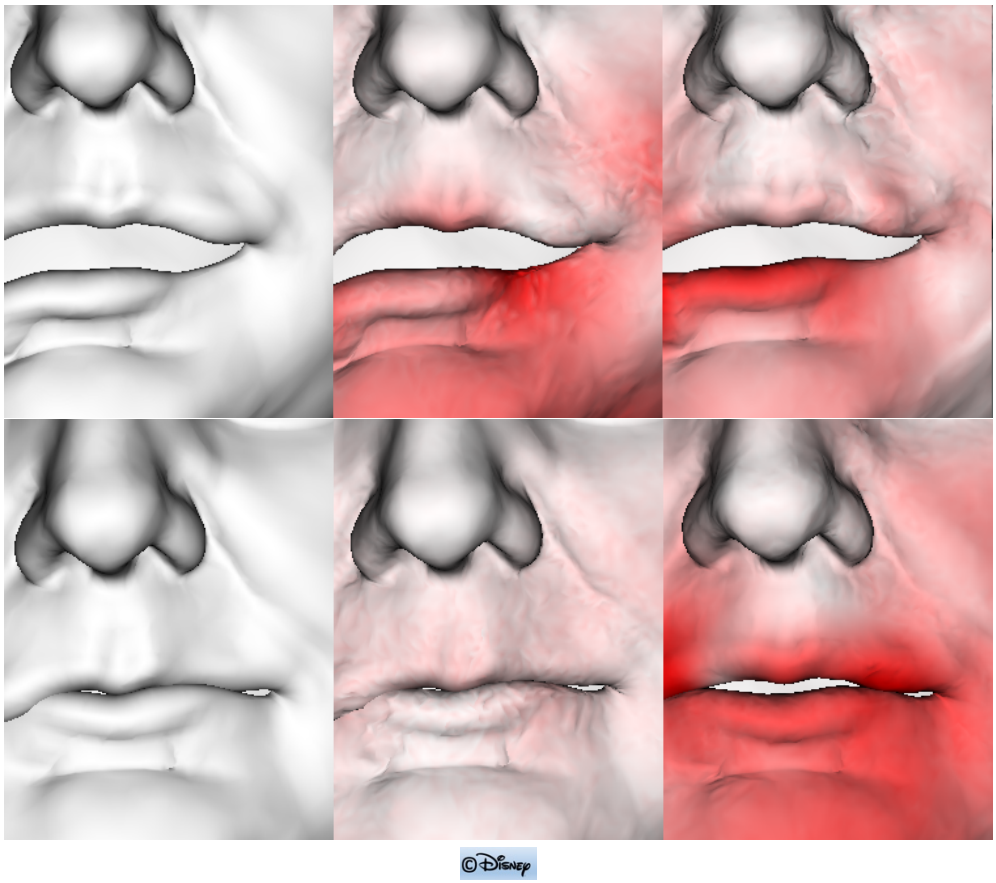



**Figure 2.** Two deep generative architectures compared on the humanoid figure (left to right) with our ground truth finite-element simulation, our KSNN architecture, and our GAN result on two poses (above and below). We generate vertex coordinates on each channel independently. The GAN uses the KSNN architecture as the generator. The discriminator consists of layered convolutions on groups of adjacent vertices (red is absolute vertex distance from simulated ground truth).

for single outlier of mechanically coupled motors) and the observed behavior of the coupled quasi-static simulator, we aligned upon a shallow, fully connected (KSNN) neural-network architecture. In our shallow network, the input-actuator parameter space is relatively small (compared to image-classification problems, for example), and we find a small number of hidden-layer neurons to yield sufficient prediction accuracy (See Figure 6). Indeed, in deeper multi-layer network tests, memory considerations for GPU solving were a factor, exceeding resources with up to 100k hidden neurons on the humanoid Audio-Animatronic® mesh. In both cases, a large number of hidden-layer neurons exhibited overfitting. We also tried more recent generative architectures such as a Generative Adversarial Network (GAN). We display a comparison of KSNN (a fully-connected architecture) to GAN (the architecture uses the same generator as KSNN and the same discriminator as the regressor in Section 6.2) in Figure 2. While there might be some trade-off in fidelity with the GAN architecture, the time to train these networks for thousands of meshes did not make it appealing for our application.

### 5.1. Actuator to Mesh Pose Generation

Our KSNN net predicts mesh-vertex components $(x, y, z)$ independently. Output values are trained relative to a regular base-mesh pose for each Audio-Animatronic® figure. We populate the actuator configurations by independently assigning each actuator a random value within an actuator-specific min-max range (unless there are pre-specified group constraints).

We generate solved mesh poses from actuator values with just three fully connected layers: an input layer, a single tanh-activated hidden layer, and a linear-activated output layer.

The input layer consists of $N$-actuator inputs, the values of which are normalized from their rotational values in degrees, to a range of [0, 1]. The size of the hidden layer is tuned to each dataset with best results generally achieved using a quantity of neurons equal to one to two times the number of input neurons (i.e., one to two times the number of actuators). For the humanoid figure, which has 13 actuators and thus 13 input neurons, a hidden layer with 13 neurons yielded the best results. For the Na'vi figure, which has 11 actuators, 22 neurons in the hidden layer yielded the best results. The size of the output layer is equal to the number of vertices in the solved mesh. Three separate networks are trained independently for the $x$, $y$, and $z$ offsets.

Training consisted of just over 5,000 simulated poses and validated on an additional 20% of unseen poses. The KSNN network was trained using an Adam optimizer with a mean-squared error-loss function, saving only the best checkpoint for each of the three networks.

## 6. Accelerated Training with Adaptive Learning

Even if the KSNN generator produces high-fidelity results as shown in Section 5, the effort it takes to produce an actuator-to-mesh dataset with the physics-based model can be prohibitively expensive in terms of simulator time.

For example, the set of possible configurations grows exponentially with the number of actuators (e.g., at least $2^{|A|}$ where $A$ is the actuator set). Also consider that there is no straightforward approach to select which actuator configurations to simulate, so the physics-based model might often hit unfeasible configurations.

It is indeed feasible to generate a useful set of poses to fit our learned models for an assembly such as the one in Figure 6 (under two days on four CPUS for 1560 poses using our physics-based model). However, the neural network might not provide a comprehensive representation of the physical models unless we identify the most difficult poses to generate, or the ones that are uncommon (to avoid an 'over-fitting' scenario).

Hence, the question is how can we guide this exploration to generate an actuator-to-mesh dataset that gives us high-fidelity results in an efficient manner.

### 6.1. Data-driven Active Learning

Motivated by the limitations outlined in Section 2 regarding the classic setup for active learning, we propose a method for actively selecting actuator-mesh pairs that makes the training process converge faster and more robustly.

Our approach is inspired by recent work in the meta-AL literature. The motivation behind meta-AL is to learn rules that do not strongly depend on intuition or ad-hoc rules (e.g., uncertainty-sampling criteria) or mathematical assumptions [Sener and Savarese 2018; Krause et al. 2008]. For example, it has been shown that uncertainty-sampling (by far the most popular approach) can behave arbitrarily bad when there is class imbalance in the initial training set [Konyushkova et al. 2017; Pelleg and Moore 2005]. Popular meta-AL approaches include [Hsu and Lin 2015; Baram et al. 2004], where a multi-armed bandit algorithm is used in conjunction with different AL sampling algorithms. Note that such an approach could only improve the results of our proposed method as it may be used in combination with other samplers. However, we leave those experiments for subsequent work as the main focus here is the novel application of LAL [Konyushkova et al. 2017] to generative modeling.

### 6.2. GEN-LAL

LAL (Learning Active Learning) [Konyushkova et al. 2017] is a framework that simulates runs of an AL method as a subroutine[2] to produce data that will help us select the incoming observations (see Figure 3). This simulation data is model state (plus

[2]Without loss of generality, the AL subroutine uses random sampling.

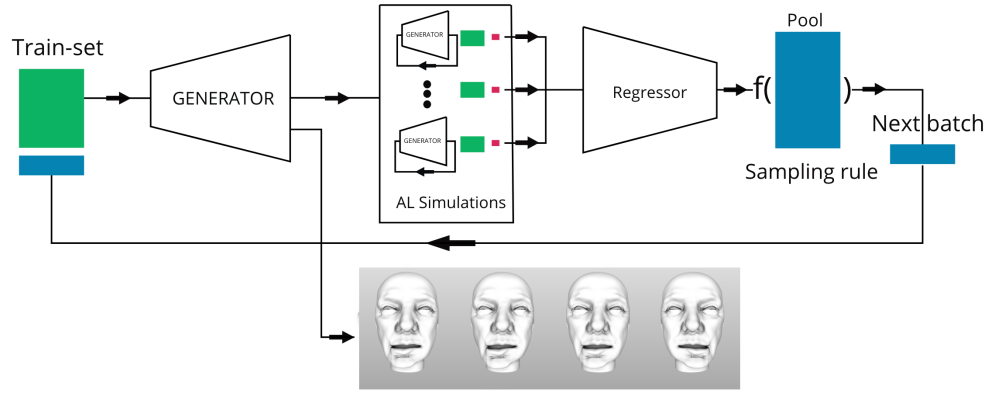





**Figure 3**. GEN-LAL (Learning Active Learning) sampling scheme. We sequentially augment the training set from a set of plausible actuator configurations. We learn the selection rule from a mapping of model-state features (predicted mesh and corresponding actuators) to yield a reduction in generalization error (see Section 6).

input observation data, e.g., actuator values) mapped to a reduction in generalization error that is computed in the test set. Once this simulated dataset is produced,[3] we train a regressor that will learn the batch selection rule (e.g., rank the observations by decreasing reduction in generalization error). Recall, *generalization error* is the measure of how accurately the model is able to predict unseen samples.

The LAL technique was initially proposed to select training pairs in the classification context. However we propose here, that as long as the model state features (in our case, the mesh that the generator yields at any point of training) can be reduced to a scalar (delta in generalization error), the framework can be successfully applied to the generation of high-dimensional observations.

For GEN-LAL, the regressor is a CNN (convolutional neural network) that reduces the mesh to a delta in generalization error. For this CNN (first layer is the $x, y, z$ channels), we used filters of varying depth $(3 \rightarrow 9 \rightarrow 18 \rightarrow 27 \rightarrow 18 \rightarrow 9 \rightarrow 3 \rightarrow 1)$ and size $(32 \rightarrow 16 \rightarrow 16 \rightarrow 8 \rightarrow 8 \rightarrow 4 \rightarrow 4 \rightarrow 4)$ while keeping a fixed stride (e.g., 2).

Additionally, we used a $0.5$ dropout regularization to train this regressor and batch normalization in the first layer to control variance.

Our contribution is the novel application of learning active learning (LAL) to the generation of unstructured meshes. The exact procedure (Section 6.2) is split into a main routine (Algorithm 1: GEN-LAL) and a sampling sub-routine (Algorithm 2:

[3]There are a few ways to produce the simulated data as outlined in [Konyushkova et al. 2017]: the simulated pairs can be independent, or we can induce a dependence which is the approach we take in Section 6.2

SIMULATED RANDOM-AL). The experiment setup and results for assessing this method can be found in Section 6.3.

**Algorithm 1: GEN-LAL Independent**

**Data:** $\{\tau_0, \ldots, \tau_N\}$ (simulation configurations).
$g$ (mesh Generator). $f$ (test error delta regressor).
$B$ sample batch size. $\mathbf{TR}$ (current train set),
$\mathbf{P}$ (observation pool), $\mathbf{TT}$ (test set)
**Result:** $\{x_1, \ldots, x_B\}$ sample batch of proposed actuator configurations
$\mathcal{D} \leftarrow \mathbf{TR}\ \mathcal{D}' \leftarrow \mathbf{TT}$;
$\mathbf{SPLIT} \leftarrow$ random partitioning function;
$\phi \leftarrow \emptyset$; $\eta \leftarrow \emptyset$; $\delta \leftarrow \emptyset$;
**for** $\tau \leftarrow \{\tau_0, \ldots, \tau_N\}$ **do**
  $(\phi_\tau, \eta_\tau, \delta_\tau) \leftarrow$ **SIMULATED RANDOM-AL**$\big(\mathcal{D}, \mathcal{D}'$,g, $\mathbf{SPLIT}, \tau\big)$ Append $(\phi_\tau, \eta_\tau, \delta_\tau)$ to $(\phi, \eta, \delta)$ respectively ;
**end**
Train regressor $f$ on $(\phi, \eta) \rightarrow \delta$;
$X_B \leftarrow \arg\max_{\mathbf{S} \subset \mathbf{P}} \sum_{x \in \mathbf{S}} f(x)$;
Return $X_B$

**Algorithm 2: Simulated Random-AL**

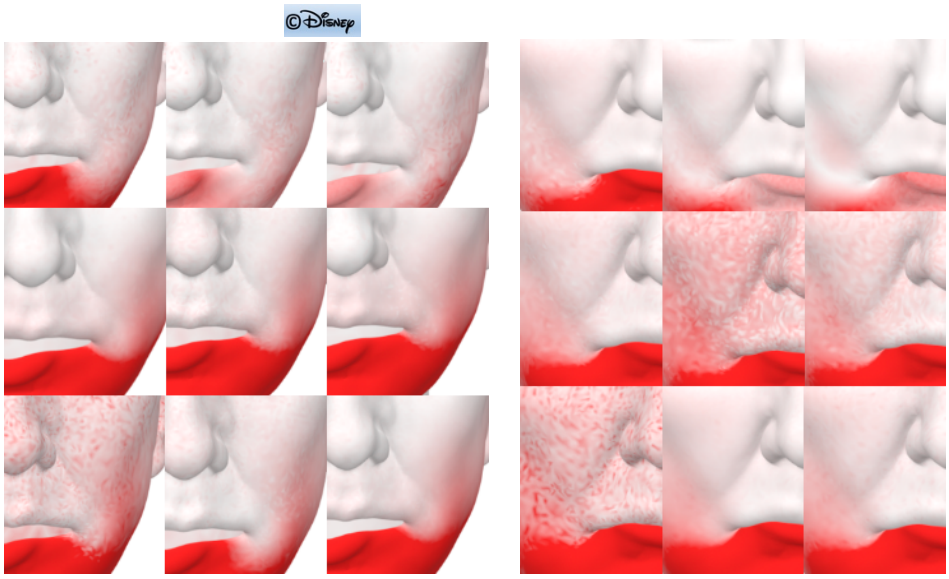
**Data:** $\mathbf{M}$ (nbr. of samples). $\mathbf{SPLIT}$ (defines init-set and simulated pool). $\mathcal{D}$ (train set). $\mathcal{D}'$ (test set). $g$ (mesh generator model)
**Result:** $(\phi, \eta)$ (model state and actuator features) and $\delta$ (reduction in generalization error) pairs
$\mathcal{L}_\tau, \mathcal{U}_\tau \leftarrow \mathbf{SPLIT}(\mathcal{D}, \tau)$;
Train generator $g_\tau$ instance on $\mathcal{L}_\tau$;
$\ell_\tau \leftarrow \ell_0$ (test set loss estimate) ;
Compute $\phi_0$ (model-state parameters) and append to $\phi$ ;
Append $\eta_0$ (actuators corresponding to $\mathcal{L}_\tau$) to $\eta$ ;
$\mathcal{L}_0 \leftarrow \mathcal{L}_\tau$;
**for** $t \leftarrow$ *1 to* $\mathbf{M}$ **do**
  Select $x_t \in \mathcal{U}_\tau$ at random;
  $\mathcal{L}_t \leftarrow \mathcal{L}_{t-1} \cup \{x\}$;
  Append $x_t$ to $\eta$;
  Train generator $g_\tau$ on $\mathcal{L}_t$;
  Compute $\phi_t$ (model-state parameters) and append to $\phi$ ;
  Append $(\ell_\tau - \ell_t)$ to $\delta$;
  $\ell_\tau \leftarrow \ell_t$ (test set loss estimate) ;
**end**
**Return** $(\phi, \eta, \delta)$

### 6.3. Experiments

Interactive editing of motor activations with real-time mesh-pose preview within a digital content-creation package (Figure 1.4) is high speed and fluid. For example, our single-thread version running on an Intel Xeon Skylake 3.2GHz CPU computes on average 1559.45µs for the humanoid figure, with 1388.44µs for the hyper-realistic figure, and 1375.81µs for the stylized Audio-Animatronic,® which has the most dense mesh with over 100k vertices solved in real time.

For our validations, we generated 5,290 actuator-to-mesh values for the humanoid head assembly (hyper-realistic figure: 5,520, stylized figure: 4,108). The actuator configurations were selected by setting each actuator to its boundary values. From all the possible controller configurations, we select them uniformly at random (choose one index independently at a time) as long as they respect the group constraints. We also remove duplicates, if any.

Using the experiment setting described above, we develop and compare GEN-LAL to a random (sample uniformly at random for each new batch) and a greedy fully informed (greedily select the batch that achieves the lowest error on the test set) baseline. Figures 4(a) and 4(b) show the L1-error of these methods as a colormap overlay upon poses that were unseen during training.




(a) (b)

**Figure 4**. Adaptive learning and partially generated poses ((a) and (b) are just different selected poses for demonstration) at 30/60/90 (left to right) percent of total training data. From top to bottom, the sampling methods are GEN-LAL, greedy fully informed, and random.

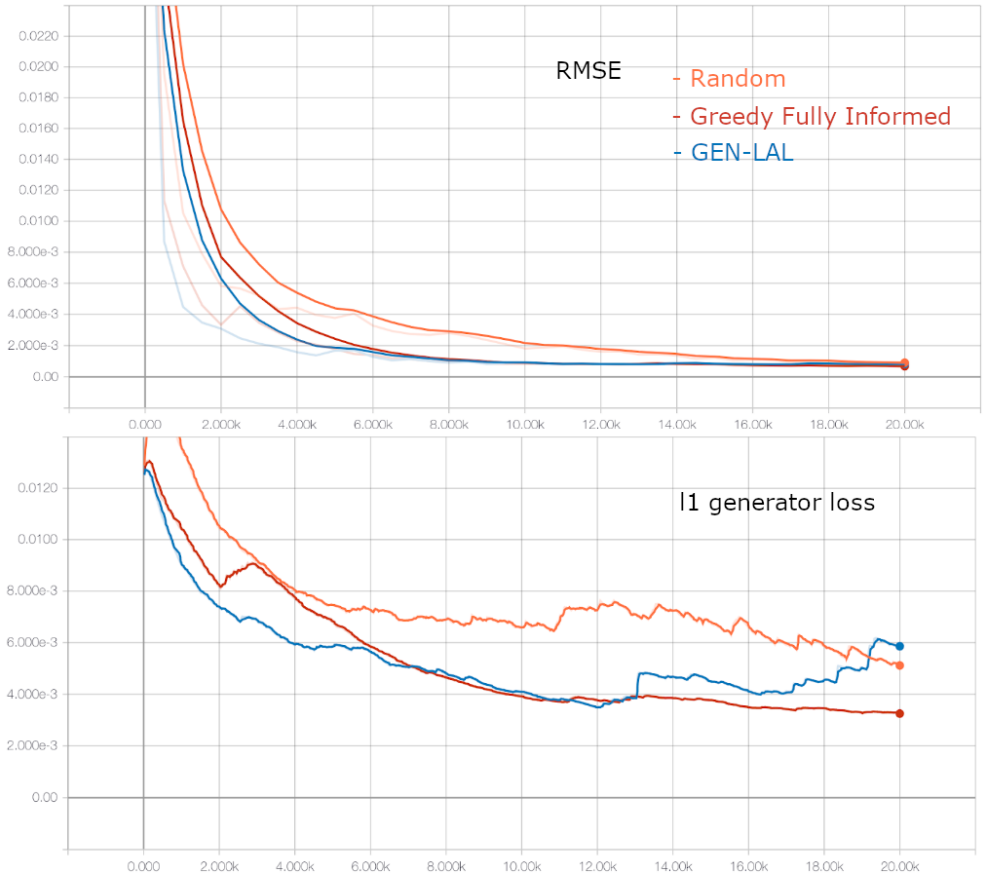



**Figure 5**. Plots of RMSE corresponding to Figure 4 on the test set and L1 parameter loss-convergence behaviors for the random, greedy fully Informed and GEN-LAL adaptive-learning sampling methods. Top: RSME; bottom: L1 generator loss (also translucent in above plot).

We observe that GEN-LAL can achieve convergence in about half the iterations required by either the Random or Fully Informed baselines (Figure 5). Moreover, we see that for regions of the face that require higher fidelity (e.g., corners of the lip and eye contours) the GEN-LAL gets much closer to the ground truth solution.

## 7. Limitations

Error is visualized in terms of surface-distance error (Figures 1 and 6) and L1-error vertex component distance error (Figures 2 and 4). The training loss function is a vertex-component distance, but may be more sophisticated. For example, Hausdorff distance or Earth Mover distance may yield a more efficient training-to-convergence process, but they may be more costly to compute, hence more time consuming to train.

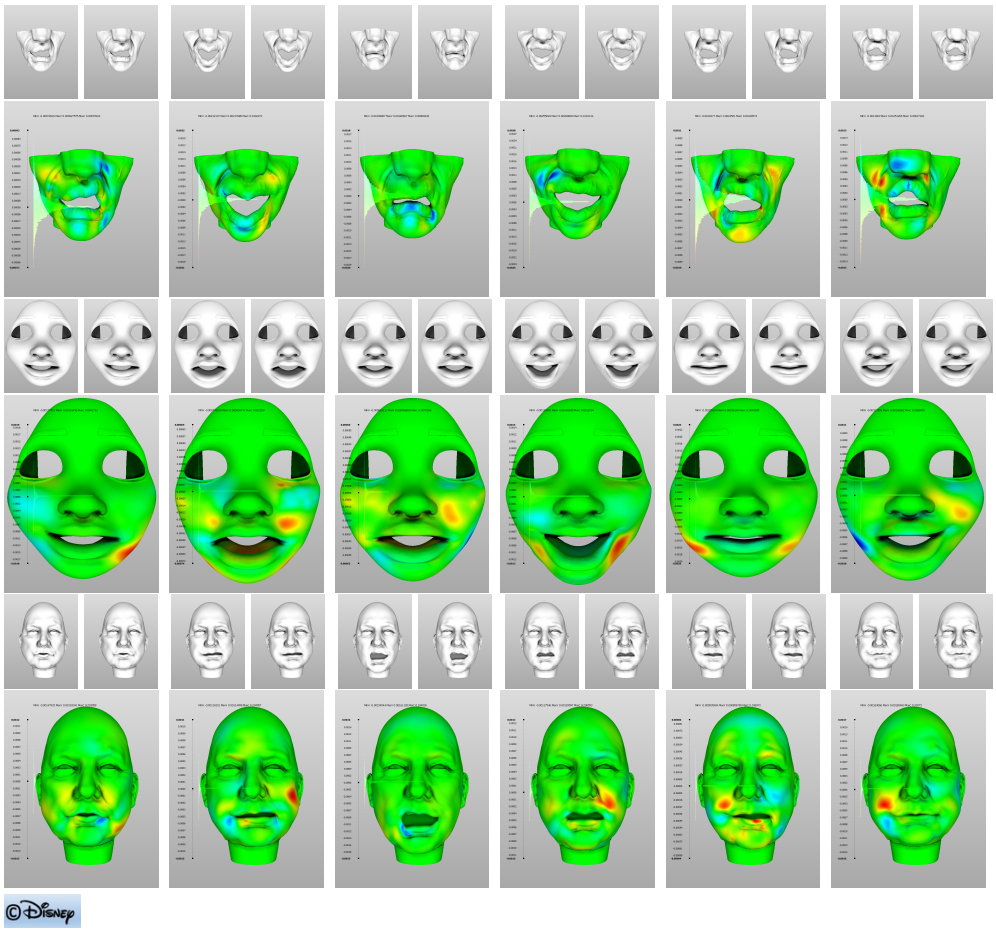



**Figure 6**. Learned Audio-Animatronic® pose-modeling results on a hyper-realistc creature, plus stylized and realistic humanoid figures. Upper rows: pairs of offline simulated ground truth and fully connected neural network (KSNN) solved-pose results; lower rows: corresponding surface-distance accuracy visualization between upper pairs, with green colors for low error, yellow/red for negative difference, and cyan/blue for positive difference. See supplementary material for more detail and further samples.

If the mechanical assemblies employed vector-motion activations, it is possible that the KSNN structure would need to combine learning of $x, y, z$ components together, resulting in longer training and solver times or less accuracy. For full face-and-body Audio-Animatronics reg a structured combination of solvers would potentially yield the best results.

Scalability of the work for very large datasets is limited by available on-board GPU memory. To utilize GPU-acceleration in the deep-learning pipeline, one or more of the following may need to be constrained:

- The quantity of simulated poses for training.

- The density of the mesh, which correlates to the number of fully-connected output neurons.
- The number of input (actuator) and hidden neurons.

The GEN-LAL approach instantiates a new simulation per sample, which incurs a computation and memory overhead in the current implementation. Variants may share initialization to make more efficient use of hardware resources. We must mention, however, that the implementation of GEN-LAL was quite taxing for GPU memory even though we used Tesla P100 hardware.[4] We imagine that running this model on larger meshes (the results of Figure 4 represent a mesh with 50 thousand vertices) would require further hardware and code optimization.

## 8. Conclusion

We provide a method for interactive Audio-Animatronic® performance design that learns to model an accurate, offline physical skin simulator adaptively for rapid incremental training and real-time solving. The result is a practical workflow for designers to digitally preview attractions without costly physical-hardware iteration cycles. The live-designed performances finally are transferred to the physical stage without loss of accuracy or engagement for guests.

In the future, we would like to investigate how to generalize the solver with a broad training set or method that can predict actuated mesh poses for any designed motor assemblies and skin configurations without training tied to each character.

While our industrial application focuses upon Audio-Animatronics,® the approaches taken may suit optimization of other complex animation deformation schemes driven by low-dimensional inputs, such as high-end feature-film production and video game animation systems.

### Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable feedback. The offline Audio-Animatronic® skin simulator used for this work includes contributions from (in alphabetic order) Moritz Geilinger, Peter Kaufmann, Espen Knoop, Max Rietmann, Gerhard Röthlin, Bernhard Thomaszewski, and Hongyi Xu.

## References

BÄCHER, M., COROS, S., AND THOMASZEWSKI, B. 2015. Linkedit: Interactive linkage editing using symbolic kinematics. *ACM Trans. Graph. 34*, 4 (July), 99:1–99:8. URL: http://doi.acm.org/10.1145/2766985. 3

[4] We implemented GEN-LAL in Tensorflow. A practical limiting factor was the pipelining of model weights for generator, simulated generator, and regressor.

BAILEY, S. W., OTTE, D., DILORENZO, P., AND O'BRIEN, J. F. 2018. Fast and deep deformation approximations. *ACM Trans. Graph. 37*, 4 (July). URL: https://doi.org/10.1145/3197517.3201300. 3

BARAM, Y., EL-YANIV, R., AND LUZ, K. 2004. Online choice of active learning algorithms. *J. Mach. Learn. Res. 5* (Dec.), 255–291. URL: https://pdfs.semanticscholar.org/f761/2bb8aba0d7c6c3c90bb82da0d9df60768217.pdf. 9

BARRIELLE, V., AND STOIBER, N. 2019. Realtime performance-driven physical simulation for facial animation. *Computer Graphics Forum 38*, 1, 151–166. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13450. 3

BECKER-ASANO, C., OGAWA, K., NISHIO, S., AND ISHIGURO, H. 2010. Exploring the uncanny valley with Geminoid HI-1 in a real-world application. In *IADIS Intl. Conf. Interfaces and Human Computer Interaction*. 01, 121–128. URL: https://www.researchgate.net/publication/229059888_Exploring_the_uncanny_valley_with_Geminoid_HI-1_in_a_real-world_application. 2

BERN, J. M., CHANG, K.-H., AND COROS, S. 2017. Interactive design of animated plushies. *ACM Trans. Graph. 36*, 4 (July), 80:1–80:11. URL: http://doi.acm.org/10.1145/3072959.3073700. 3

BICKEL, B., KAUFMANN, P., SKOURAS, M., THOMASZEWSKI, B., BRADLEY, D., BEELER, T., JACKSON, P., MARSCHNER, S., MATUSIK, W., AND GROSS, M. 2012. Physical face cloning. *ACM Trans. Graph. 31*, 4 (July), 118:1–118:10. URL: http://doi.acm.org/10.1145/2185520.2185614. 3

BLITZ, M. 2019. The a1000 is disney's advanced animatroinc bringing star wars: Galaxy's edge to life. *Popular Mechanics* (Feb.). 2

CEYLAN, D., LI, W., MITRA, N. J., AGRAWALA, M., AND PAULY, M. 2013. Designing and fabricating mechanical automata from mocap sequences. *ACM Trans. Graph. 32*, 6 (Nov.), 186:1–186:11. URL: http://doi.acm.org/10.1145/2508363.2508400, doi:10.1145/2508363.2508400. 3

COROS, S., THOMASZEWSKI, B., NORIS, G., SUEDA, S., FORBERG, M., SUMNER, R. W., MATUSIK, W., AND BICKEL, B. 2013. Computational design of mechanical characters. *ACM Trans. Graph. 32*, 4 (July), 83:1–83:12. URL: http://doi.acm.org/10.1145/2461912.2461953. 3, 5

FANELLI, G., DANTONE, M., GALL, J., FOSSATI, A., AND VAN GOOL, L. 2013. Random forests for real time 3D face analysis. *International Journal of Computer Vision 101*, 3 (Feb), 437–458. URL: https://doi.org/10.1007/s11263-012-0549-0. 7

GEILINGER, M., PORANNE, R., DESAI, R., THOMASZEWSKI, B., AND COROS, S. 2018. Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels. *ACM Trans. Graph. 37*, 4 (July), 160:1–160:12. URL: http://doi.acm.org/10.1145/3197517.3201368. 3

GLUCK, K. 2013. The early days of audio-animatronics. *The Walt Disney Family Museum Blog*. URL: https://www.waltdisney.org/blog. 2

HOLT, G. 2017. Blurring the mechanical line: GHP's new animatronic head offers a glimpse into the future of expressive characters. In *This Animatronic Life*. MiceChat. URL: https://www.micechat.com. 2

HSU, W.-N., AND LIN, H.-T. 2015. Active learning by learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, B. Bonet and S. Koenig, Eds. AAAI Press, Palo Alto, CA, UsA, 2659–2665. URL: http://dblp.uni-trier.de/db/conf/aaai/aaai2015.html#HsuL15. 9

ICHIM, A.-E., KADLEČEK, P., KAVAN, L., AND PAULY, M. 2017. Phace: Physics-based face modeling and animation. *ACM Trans. Graph. 36*, 4 (July), 153:1–153:14. URL: http://doi.acm.org/10.1145/3072959.3073664. 3

KAZEMI, V., AND SULLIVAN, J. 2014. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, USA, 1867–1874. 7

KONYUSHKOVA, K., RAPHAEL, S., AND FUA, P. 2017. Learning active learning from data. In *Conference on Neural Information Processing Systems*, Neural Information Processing Systems Foundation, San Diego, CA, USA. URL: https://papers.nips.cc/paper/7010-learning-active-learning-from-data.pdf. 9, 10

KRAUSE, A., SINGH, A., AND GUESTRIN, C. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research (JMLR) 9* (February), 235–284. URL: https://www.jmlr.org/papers/volume9/krause08a/krause08a.pdf. 4, 9

MEGARO, V., THOMASZEWSKI, B., NITTI, M., HILLIGES, O., GROSS, M., AND COROS, S. 2015. Interactive design of 3d-printable robotic creatures. *ACM Trans. Graph. 34*, 6 (Oct.), 216:1–216:9. URL: http://doi.acm.org/10.1145/2816795.2818137. 3

MEGARO, V., ZEHNDER, J., BÄCHER, M., COROS, S., GROSS, M., AND THOMASZEWSKI, B. 2017. A computational design tool for compliant mechanisms. *ACM Trans. Graph. 36*, 4 (July), 82:1–82:12. URL: http://doi.acm.org/10.1145/3072959.3073636. 3

MORI, M. 1970. The uncanny valley (in japanese). *Energy 7*, 33–35. English version published in *IEEE Robotics and Information Magazine*. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6213238. 2

NOCEDAL, J., AND WRIGHT, S. 2006. *Numerical optimization*, 2. ed. Springer series in operations research and financial engineering. Springer, New York, NY. URL: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+502988711&sourceid=fbw_bibsonomy. 6

PELLEG, D., AND MOORE, A. W. 2005. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, Cambridge, MA, USA, 1073–1080. URL: http://papers.nips.cc/paper/2554-active-learning-for-anomaly-and-rare-category-detection.pdf. 9

SENER, O., AND SAVARESE, S. 2018. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of ICLR*, International Conference on Learning Representations, La Jolla, CA, USA. URL: https://openreview.net/forum?id=H1aIuk-RW. 4, 9

SETTLES, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. URL: http://burrsettles.com/pub/settles.activelearning.pdf. 4

SIFAKIS, E., AND BARBIC, J. 2012. Fem simulation of 3d deformable solids: A practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, ACM, New York, NY, USA, SIGGRAPH '12, 20:1–20:50. URL: http://doi.acm.org/10.1145/2343483.2343501. 6

SIVARAMAN, S., AND TRIVEDI, M. 2014. Active learning for on-road vehicle detection: a comparative study. URL: https://doi.org/10.1007/s00138-011-0388-y. 4

SKOURAS, M., THOMASZEWSKI, B., COROS, S., BICKEL, B., AND GROSS, M. 2013. Computational design of actuated deformable characters. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 32*, 4, 82–1–82–10. URL: https://cgl.ethz.ch/publications/papers/paperSko13.php. 3

THOMASZEWSKI, B., COROS, S., GAUGE, D., MEGARO, V., GRINSPUN, E., AND GROSS, M. 2014. Computational design of linkage-based characters. *ACM Trans. Graph. 33*, 4 (July), 64:1–64:9. URL: http://doi.acm.org/10.1145/2601097.2601143. 3

TONG, S., AND KOLLER, D. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res. 2* (Mar.), 45–66. URL: https://doi.org/10.1162/153244302760185243. 4

WARMUTH, M., LIAO, J., RÄTSCH, G., MATHIESON, M., PUTTA, S., AND LEMMEN, C. 2003. Support vector machines for active learning in the drug discovery process. *J Chem Inf Comput Sci 43*, 2 (Mar–Apr), 667–673. URL: https://pubmed.ncbi.nlm.nih.gov/12653536/. 4

ZEHNDER, J., KNOOP, E., BÄCHER, M., AND THOMASZEWSKI, B. 2017. Metasilicone: Design and fabrication of composite silicone with desired mechanical properties. *ACM Trans. Graph. 36*, 6 (Nov.), 240:1–240:13. URL: http://doi.acm.org/10.1145/3130800.3130881. 3

ZHU, L., XU, W., SNYDER, J., LIU, Y., WANG, G., AND GUO, B. 2012. Motion-guided mechanical toy modeling. *ACM Trans. Graph. 31*, 6 (Nov.), 127:1–127:10. URL: http://doi.acm.org/10.1145/2366145.2366146. 3

## Index of Supplemental Materials

Supplementry materials for this paper include additional results (jcgt.org/published/0009/03/01/supplementary_results.pdf) as well as a video demonstrating interactive pose editing with an artist-friendly digital content-creation package plugin, feeding into the fully realized Audio-Animatronic® (jcgt.org/published/0009/03/01/video.mp4).

## Author Contact Information

Joel Castellon
Disney Research Los Angeles
521 Circle Seven Drive
Glendale, CA, 91201
joeldancastellon@gmail.com

Matt McCrory
Wat Disney Imagineering
521 Circle Seven Drive
Glendale, CA, 91201
Matt.McCrory@gmail.com

Moritz Bächer
Disney Research Los Angeles
521 Circle Seven Drive
Glendale, CA, 91201
moritz.baecher@disneyresearch.com

Alfredo Ayala
Wat Disney Imagineering
521 Circle Seven Drive
Glendale, CA, 91201
Alfredo.M.Ayala@disney.com

Jeremy Stolarz
Wat Disney Imagineering
521 Circle Seven Drive
Glendale, CA, 91201
Jeremy.Stolarz@disney.com

Kenny Mitchell
Disney Research Los Angeles
and Edinburgh Napier University
521 Circle Seven Drive
Glendale, CA, 91201
k.mitchell2@napier.ac.uk

Joel Castellon, Matt McCrory, Moritz Bächer, Alfredo Ayala, Jeremy Stolarz and Kenny Mitchell, Active Learning for Interactive Audio-Animatronic® Performance Design, *Journal of Computer Graphics Techniques (JCGT)*, vol. 9, no. 3, 1–19, 2020
http://jcgt.org/published/0009/03/01/

| | | | |
|---|---|---|---|
| Received: | 2019-10-31 | | |
| Recommended: | 2020-03-12 | Corresponding Editor: | Joe Geigel |
| Published: | 2020-10-11 | Editor-in-Chief: | Marc Olano |