Doctoral Dissertation

Data-Driven Haptic Modeling and Rendering of Viscoelastic Behavior using Fractional Derivatives

Hojun Cha (차 호 준)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2022

분수계 미분을 활용한 데이터 기반 점탄성 촉감 모델링 및 렌더링

Data-Driven Haptic Modeling and Rendering of Viscoelastic Behavior using Fractional Derivatives

Data-Driven Haptic Modeling and Rendering of Viscoelastic Behavior using Fractional Derivatives

by

Hojun Cha
Department of Computer Science and Engineering
Pohang University of Science and Technology

A dissertation submitted to the faculty of the Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Computer Science and Engineering

Pohang, Korea
12. 27. 2022
Approved by
Seungmoon Choi (Signature)
Academic advisor

Data-Driven Haptic Modeling and Rendering of Viscoelastic Behavior using Fractional Derivatives

Hojun Cha

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a doctoral degree from POSTECH

12. 27. 2022

Committee Chair	Seungmoon Choi	(Seal)
Member	Inseok Hwang	(Seal)
Member	Keehoon Kim	(Seal)
Member	Minsu Cho	(Seal)
Member	Seokhee Jeon	(Seal)

Data-Driven Haptic Modeling and Rendering of Viscoelastic Behavior using Fractional Derivatives

Hojun Cha

The undersigned have examined this dissertation and hereby certify that it is worthy of acceptance for a doctoral degree from POSTECH

12. 27. 2022

Committee Chair	Seungmoon Choi	(Seal)
Member	Inseok Hwang	(Seal)
Member	Keehoon Kim	(Seal)
Member	Minsu Cho	(Seal)
Member	Seokhee Jeon	(Seal)

DCSE 차 호 준 Hojun Cha
20142825 Data-Driven Haptic Modeling and Rendering of Viscoelastic Behavior using Fractional Derivatives.
분수계 미분을 활용한 데이터 기반 점탄성 촉감 모델링 및 렌더링.
Department of Computer Science and Engineering , 2022, 74p
Advisor : Seungmoon Choi.
Text in English.

ABSTRACT

We interact with many objects with a touch, a natural exploratory procedure. Similar to the real world, modeling and rendering an interactive haptic model of an object in the virtual world has been done to give immersiveness and realism. A viscoelastic behavior is one of the target models, and it has been trained using a data-driven method due to the non-linearity. The modeling was successful, except that it becomes computationally intractable in a large dataset.

In this dissertation, we propose a fractional derivative as a modeling feature on a data-driven haptic modeling and rendering method of viscoelastic deformable objects. Here, we propose an optimized form of the Grünwald–Letnikov derivative for haptic modeling and rendering, considering the time quantization and double precision in computing. We compute the fractional derivatives from the zeroth to the first order to get richer information.

Modeling and rendering the normal interaction of a homogeneous object is performed using fractional derivatives. The discrete-time interaction data (position and force) is collected for five viscoelastic deformable objects. Regression forest models (RF-FD) and radial basis function interpolation models (RBF-FD) are trained on the features of the fractional derivatives as input-output mapping functions. The results are also compared with the RBF-based data-driven approach (RBF-PV). It validates that the features based on fractional derivatives are better than the standalone position and velocity information for modeling deformable objects. Then, We employ the trained RF models for haptic rendering and compare subjective similarity between the rendered and real objects through human perceptual experiments. Results show that the models trained with fractional derivatives show higher realism than the RBF model trained with position and velocity information. Most of the participants could not discriminate between real and virtual objects.

In the later part of this dissertation, we further optimize this modeling method using the perceptual sampling method. This method is based on the perceptual threshold of the force difference, which is about 5%. In the modeling results, the perceptual sampled models show dramatically low modeling times and relatively low error rates compared to the random sampling method. The human experiment of comparison between the random and the perceptual sampling revealed that our proposed sampling method marked a statistically worse score than the previous one. However, these similarity scores have a possibility to improve rendering quality, such as by providing stable rendering or complex input signals.

Contents

List of Tables

List of Figures

I. Introduction

1.1 Research Motivation

Human, social being, wants to be connected. We build connections with our family, friends, colleagues, and other people in the real world. Physical space is, however, not enough to keep every connection. Humankind develops technology to break this limitation using every communication method. Nowadays, we meet our parents or family far away using video calls, have fun with friends through 3D game avatars, or share our knowledge with colleagues at a virtual conference. Even though we see their face and hear their voice, it does not provide exact emotional feelings and atmospheres at the real meeting due to a lack of a haptic feeling. And every haptic research question originates from here.

The haptic sensation can deliver rich information in an interactive scenario compared to vision and audition. A human visual system model is good enough to estimate objects with passively-given 2-D static images. However, we can only estimate its surface properties for haptic sensation in a static contact, such as its hardness or temperature. It is because haptic sensations focus more on active touch than passive touch. This nature leads the haptic researchers to build interactive models rather than record and playback the haptic properties.

Assume we interact with an object, such as a soft doll, with our finger. We can slightly touch and slip the object's surface to figure out its position and shape, rub it left and right to feel the textures, and gently push it to estimate its material properties. All these haptic sensations from the interactions are combined in our brains to build a perceptual model of one object. These information acquisition processes are called exploratory procedures (EP) [1]. When we do passive and active touches, the object provides its shape, texture, volumetric information, and material properties. To give a presence of haptic sensation, we consider these haptic properties.

A large amount of research has been done to model haptic sensation. These works are categorized into (1) physical equation-based models and (2) data-driven models. The former models are supported by well-defined physical equations. For instance, we can easily build interaction with a solid object because its movement is controlled in a simple calculation. This method has the best performance on modeling and rendering the linear properties. However, for the non-linear properties such as viscoelasticity, it is only successful on the narrow range of interactions. To deal with the complex properties, the researchers have used the non-linear mapping functions that include input features and output resultant haptic properties, so-called data-driven haptic models. Data-driven modeling and rendering viscoelasticity is successful using a radial-basis function (RBF) except that it becomes computationally intractable in a large dataset.

We address this problem with new model features and a new modeling method. We first attempted to model the viscoelasticity using the position and the past positions trained on the random forest [2]. The results were successful in the modeling, but it became easily unstable in rendering due to the negative damping. The main issue is that the random forests with past positions cannot perfectly predict whether the user's hand is going up or down.

In this dissertation, we first explore the possibility of the fractional derivative as a feature of viscoelasticity in data-driven haptic modeling and rendering. We hope our report on the fractional derivatives in haptic modeling and rendering inspires haptic researchers who use time features such as velocity or acceleration.

1.2 Organization

This dissertation is organized as follows. Chapter II shows the works for modeling viscoelasticity, especially fractional derivatives, and the current states of data-driven haptics. It also includes our last modeling results on the features of the current and past samples. In Chapter III, we introduce our numerical background of fractional derivatives. This method, derivated from Grünwald–Letnikov derivatives, is optimized for numerical calculation.

Chapter IV introduces the modeling and rendering results using fractional derivatives. In Section 4.2, the experimental setup for data collection is explained. The following section defines the fractional derivatives modeling methods and validates the modeling results using the comparison with the previous state-of-the-art method. In Section 4.4, the trained models are employed for haptic rendering. Results of human subjective experiments are also discussed here.

The following chapter, Chapter V, describes the perceptual sampling method to improve the fractional derivatives method. We further train the same models with the sampling method and evaluate them with the previous models. Section 5.2 shows the performance compared with the previous models in error rates, and Section 5.3 validates the modeling results with a human experiment. Finally, the paper is concluded in Section VI.

II. Related Works

2.1 Viscoelastic behavior

Viscoelasticity is one of the most typical haptic behavior. We can easily find viscoelastic material in biological materials like human skin tissue [3]. There have been tremendous studies and applications to model and represent it in the virtual world, such as a surgery simulation or a human interaction scenario. To this end, the early pioneers tried to describe the complex viscoelastic behavior based on mechanical elements such as the Hookean spring and Newtonian damper. One of the basic equations for viscoelasticity is as follows:

$$F = kx + b\dot{x} \tag{2.1}$$

where x is the position, $\dot{x}$ is the velocity, k is the spring coefficient, b is the damping coefficient, and F is the response force of the material. In equation 2.1, viscoelasticity is represented as a parallel connection of a spring and a damper, and it is called the Kelvin-Voigt model. Alternatively, the serial connection of two elements, named Maxwell model, was also studied. These equations represent the relationship between the response force and the main physics terms, position and velocity. The studies had been performed to fit the coefficients of the equations to the recording of the real object, and expand the models to the generalized form using a combination such as tje generalized Maxwell model (GMM) [4]. Also, an attempt to model it using finite element method models was also proposed [5]. This series of studies had been widely spread due to its simplicity. However, these models have the limitation that they are only accurate in limited conditions.

2.2 Fractional derivatives

The study of the viscoelastic behavior continued through the fractional derivatives (FD). In the literature, Scott-Blair and Coppen proposed an alternative method to describe viscoelastic behavior based on fractional derivatives. First, they defined the “firmness” using a unified equation for the intermediate status of elasticity and viscosity [6]. Throughout the series of studies [7, 8, 9], their idea suggested a new rheological model named springpot using a fractional differential equation on stress [10]. They described this as a quasi-property.

Many attempts combined the springpot with other traditional mechanical elements to describe complex behavior for many-real world objects. The fractional calculus reduced the number of coefficients while modeling the stress-strain relationship of viscoelastic deformable objects [11, 12]. Many researchers have tried to adopt this concept to model the viscoelastic behaviors of tissues [13, 14], linear viscoelastic behaviors of elastomers [15, 16], or build a governing equation of some materials [11, 17].

In the area of haptic modeling and rendering, using fractional derivatives is one of the novel topics. Recently, there have also been efforts to employ fractional derivative components in haptic rendering models using springpot elements. In [18], the authors studied the perceptual meaning of the springpot in the rendering model. They describe the springpot using the amplitude and the phase in the frequency domain, then investigate how the firmness and the bounciness in perceptual feeling change as the parameters change. Their paper first investigates the perception of springpots using familiar physical concepts. In haptic texture modeling and rendering, Hu and Song [19] used adaptive fractional differential method to extract texture features from the image.

2.3 Data-driven modeling

The recent advances in machine learning have allowed data-driven modeling to be actively used for describing complex phenomena in many fields of science and engineering, such as physical modeling, material modeling, weather forecasting, and financial forecasting [20]. In physics, researchers have developed physics-informed machine learning models trained on the additional information more than the physical laws [21]. For example, Hatfield et al. [22] has highlighted the importance of data-driven methods for high-energy-density physics. In [23], the authors have used deep learning frameworks to solve non-linear partial differential equations. Cenedese et al. [24] has developed a data-driven modeling and prediction method for non-linear dynamic systems. Researchers have also started employing data-driven methods in material science [25]. Forecasting or predicting is also one of the promising topics in data-driven modeling [26]. For example, data-driven methods are employed for predicting the electricity consumption of a building [27] and wildfire forecasting [28].

2.4 Data-driven haptics

This dissertation lies in the broad domain of data-driven haptics. Data-driven haptics consists of three different steps: (1) data collection, (2) haptic modeling, and (3) haptic rendering. We first measure the target interaction data between the measurement setup and the object. Then, we train some models from the extracted features of the collected data. Finally, we render the interaction, such as the response force, using the rendering setup. The most different step from the physical equation-based modeling is the modeling part. As mentioned above, the physics-based model uses some form of the physics-based equation to fit the interaction force. The data-driven haptics learns a non-parametric input-output mapping function or a complex neural network framework.

The first idea started from the paper that suggested “Haptic Camera” [29]. This paper successfully trained piecewise linear dynamic models of the 1-D interaction force of a toggle switch using a measured position and position. In another study [30, 31], authors record cutting forces from surgical tissues and then tune these signals via piecewise linear models to provide force feedback. Similarly, in [32], a polynomial function is employed to model recorded force signals for a pin insertion task. Also, in [33], the authors apply a non-linear model to the recorded force and acceleration signals. The obtained parameters are then used for providing force feedback. The other related work utilizes the recorded data to acquire surface properties, such as texture or friction. For example, in [34], authors obtain properties like texture and compliance, and in [35], parameters of the friction model are estimated from the recorded data. In another study [36], authors scan 3D surfaces and estimate surface compliance based on force-acceleration profiles. As mentioned above, this model-free method employs non-parametric machine learning techniques to learn input-output mapping functions for modeling the complex behavior of objects in a virtual environment. Apart from modeling deformable objects, the approach has also been successfully employed in texture rendering [37, 38, 39] and thermal rendering [40]. It also expanded to comprehensive modeling and rendering using multiple sensors, including shape, texture mapping, deformation, and sound [41, 42].

In [43], the authors first proposed a data-driven model for modeling viscoelastic deformable objects. They employed radial basis functions (RBF) based interpolation method for learning input-output mapping function. The approach considered position, velocity, and filtered velocity as input features and the response force as the output. The RBF-based model successfully predicted the response force (output). Later, the approach was extended for modeling slip and inhomogeneous interactions on deformable objects [44, 45]. Yim et al. successfully employed the RBF-based method for modeling frictional responses during sliding and sticking interactions on inhomogeneous deformable objects [46]. It has been found that the RBF-based model, when trained on large datasets, becomes computationally intractable. However, recently in [47], a feature-based learning scheme has been combined along with the RBF-based

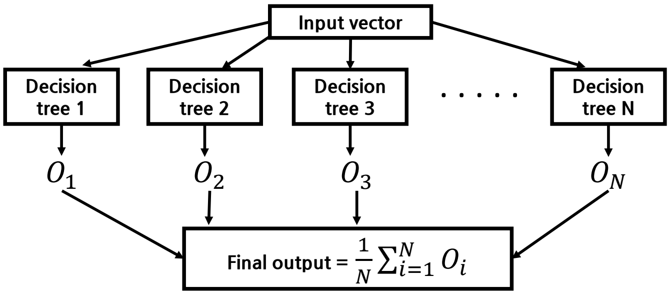


Figure 2.1: Random forest regression algorithm.

model to reduce the dimensionality of the dataset.

A very recent paper proposed the FEM-based models for the realistic modeling and rendering of viscoelasticity in real time [48]. The authors molded the object of hyper-elastic materials and added markers at the surface. They tracked the markers using external cameras to capture the relationship between the force and the resultant deformation when the force is given to the object. Then, they built a real-time predictor using FEM models. In their recent paper, they also improve the model using the inverse reinforcement learning method to train the plasticity [49]. These papers are the most complicated way to model and render with a benefit from the computing power provided that the modeling environment is set and enough data is provided.

2.5 Random Forests

In this work, we use the random forest for regression, not for classification. A random forest consists of many decision trees. A decision tree is a hierarchical structure defined by nodes, edges, and leaves. Edges connect the nodes or leaves, and leaves determine the outcome of the tree. A regression tree minimizes the mean-squared error (MSE) of the values (here, “value” refers to the dependent variable, i.e., response force) at each internal node. This leads to further splitting of the tree. Thus, while growing the tree, the MSE of the values is minimized in each subset until a stopping criterion is reached. For the regression tree, the final outcome is the average of all instances present in the leaf node. Unlike a standard decision tree, a decision tree in the random forest is trained on a subset of all the observations. For example, if there are N observations in the original data, a random subset of N' observations is chosen from the original data with replacement (this process is called 'Bagging'). The chosen subset is called the bootstrap sample. Each decision tree in the random forest is trained on a different bootstrap sample. In addition, unlike the standard decision tree, the tree in the random forest chooses the best attributes among a subset of predictors, not among all. These steps reduce the over-fitting problem. Each decision tree in the forest is learned independently of each other, and the final outcome of the random forest for a regression problem is the average of all the trees, as shown in Fig. 2.4. Apart from this, the random forest is capable of handling a large number of input attributes (both numerical and categorical), and it has many other advantages like handling of missing values and outliers. For its implementation, we have used the source code given in [50].

2.6 Previous work on the past position samples

In one of our previous works, we employed the random forest for modeling deformable viscoelastic objects [2]. This method builds a relationship between the response force and the input feature vector consisting of current and past position samples. It was observed that the RF-based model outperformed the RBF-based model in terms of prediction accuracy, training data, and computational time.

We employed our trained models in the rendering setup. The trained models were loaded in the virtual environment, CHAI3D, which is implemented on a well-known haptic rendering framework. We place a static cube that matches the real object in the rendering. Then, we keep tracking the current position to calculate the distance between the cube's surface and the haptic interaction point (HIP). The 100 ms of the tracked position data is stacked as a queue to be reshaped into a feature vector. The random forests traverse the tree with the input vector for each tree. We use the regression on the results.

At this point, we faced instability issues from the trained models, which was critical because it changed our research direction. A phantomic movement, an unwanted large force against a user's hand direction, frequently appeared during the interaction with the trained model. It was even severe when the user kept staying in the middle of the interaction. Any noise-canceling methods were not effective in this inherent problem.

We decomposed the forces from the model using a simple model of a spring and a damper to compare it with the ground-truth interaction force. We found that the damping term sometimes became *negative*, which was critical to the stability. This unstable model was compared with a stable random forest model trained with the feature of position and velocity. We conclude that the model that uses position and ten past positions sometimes does not imply the velocity term.

2.7 Sampling method

In the literature on haptic data communication [51, 52, 53, 54], a perceptually adaptive sampling mechanism has been used to reduce the data. The mechanism is based on the Weber's law of perception, and selects only perceptually significant samples for the transmission. The perceptual significance is determined by the Weber fraction δ. The approach reduces the data upto $90 - 95\%$. The mechanism is named as perceptual deadband/deadzone in the literature.

III. Computational Fractional Derivatives

A fractional derivative, first proposed in 1695 by L'Hospital, is a derivative of arbitrary order, real or complex. As with integer-order derivatives, the r^{th}-order fractional derivative of a function $f(x)$ is described as: 3.1.

$$D^r f(x) = \frac{\mathrm{d}^r f(x)}{\mathrm{d}x^r}, \tag{3.1}$$

where D^r represents the operator for the r^{th}-order fractional differentiation.

In this dissertation, we implement one of the fractional derivatives methods, such as Grünwald–Letnikov, Riemmann-Liouville, or Caputo fractional derivatives [55]. Our numerical calculation method of fractional derivatives starts from the Grünwald–Letnikov derivative for computation efficiency. This approach extends the classical definition of derivation. From the generalized form of n-th integer order derivatives, the equation can be described using binomial coefficients like the below equation.

$$f^{(n)}(x) = \lim_{h\to 0} \frac{\sum\limits_{0\le m\le n} (-1)^m \binom{n}{m} f(x+(n-m)h)}{h^n} \tag{3.2}$$

Here, h is the time difference, n is the fractional order, m is the order for the polynomial equation, and $\binom{n}{m}$ is the binomial coefficient. The binomial coefficient is generally represented upon the Γ function when we remove the limitation that n is a positive integer. Therefore, we rewrite Eq. 3.2 like the below equation.

$$\begin{aligned} D^r f(x) &= \lim_{h\to 0} \lim_{m\to\infty} \sum_{k=0}^{m} c(r,k) f(x+(r-k)h), \\ c(r,k) &= \frac{(-1)^k}{h^r} \frac{\Gamma(r+1)}{\Gamma(k+1)\Gamma(r-k+1)} \end{aligned} \tag{3.3}$$

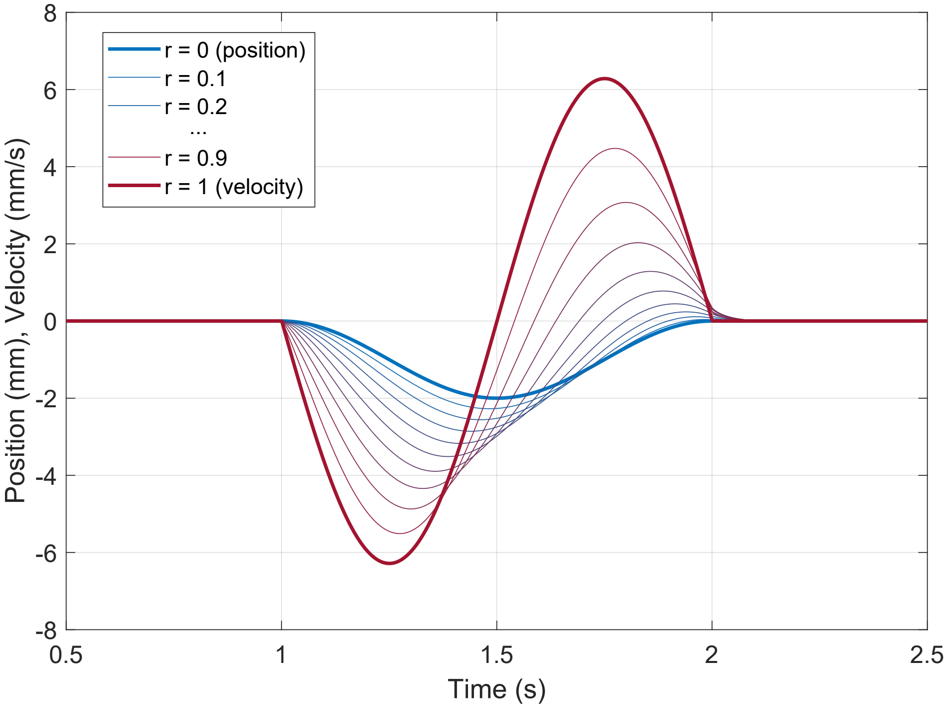


Figure 3.1: An example of fractional derivatives by its orders. It show the displacement and its fractional-order derivatives from 0.1 to 1.

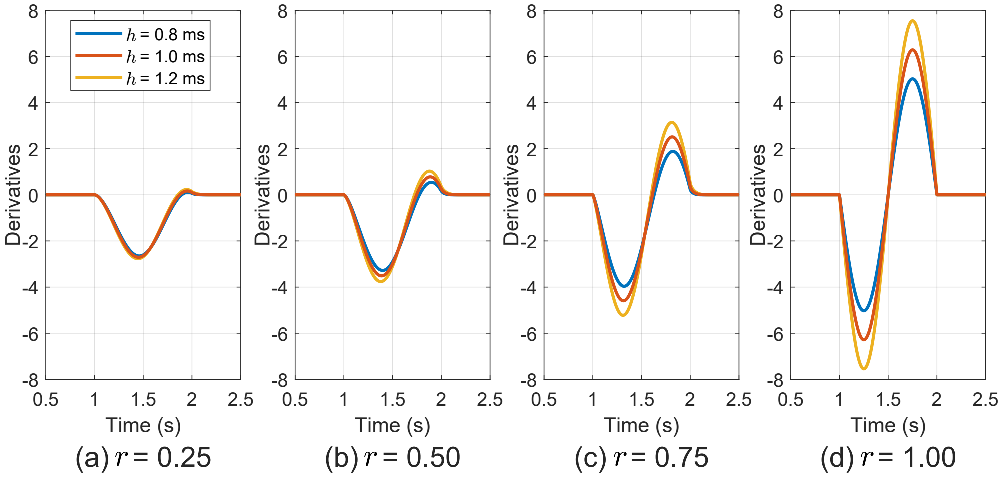
In our modeling and rendering processes, the independent variable is the time t, and we approximate (3.3) by taking small h and large m, such that

$$\begin{aligned} D^r f(t) &= \sum_{k=0}^{m} c(r,k) f(t + (r-k)h) \\ c(r,k) &= \frac{(-1)^k}{h^r} \frac{\Gamma(r+1)}{\Gamma(k+1)\Gamma(r-k+1)}. \end{aligned} \tag{3.4}$$

We optimize Eq. 3.3 for our modeling and rendering purposes. In haptic rendering, we quantize analog signals such as time or position. Our modeling and rendering update rate can be set to 1,000 Hz, which means the time difference h is 1 ms. The order of the polynomial equation m is also taken as the maximum within the floating point number range. We use $m = 171$ as larger values cause overflow in computing the gamma functions (double floating-point type, 64-bit CPU). Given r, the coefficients $c(r,k)$ are precomputed for all k $(0 \leq k \leq m)$. Then, (3.4) works as a filter on $f(t)$ and can be easily computed in real-time.

One exemplar calculation result of these fractional derivatives is shown in Fig. 3.1. It describes the 1 Hz frequency signal and ten orders of fractional derivatives from it. The fractional derivatives show the gradual transition between the position, $f(t) = D^0 f(t)$, and the velocity, $f'(t) = D^1 f(t)$.

During haptic rendering, the update rate may vary depending on many internal and external factors. Figure 3.2 shows changes in the computed values of fractional derivatives for different sampling times. Compared to the reference at $h = 1$ ms, the values remain very similar for the shorter (0.8 ms) and longer (1.2 ms) sampling time when the derivative order r is low until $r = 0.50$. The differences become more notable as r increases further to 1.00.



(a) r = 0.25 (b) r = 0.50 (c) r = 0.75 (d) r = 1.00

Figure 3.2: Effects of different sampling intervals on the computed values of fractional derivatives.

IV. Modeling and Rendering of Homogeneous Behavior

In this chapter, we train normal interaction models on the viscoelastic homogeneous deformable objects (i.e. 1-D interaction). We introduce the modeling and rendering method using the fractional derivatives on random forests and radial basis functions, which is our main approach.

4.1 Overview

The data-driven modeling and rendering method usually consists of data colleciton, modeling, and rendering. This chapter firmly follows the general flow of the data-driven research.

First, we choose silicone materials which shows viscoelastic behavior. These materials are easy to mold in any kinds of shape and also deformable and shape-restorable. We build cubic objects to standardize them. Then, we design that the input-output function using the concept of the fractional derivatives. This concept of the fractional derivatives extends the previous approaches of position and velocity pairs towards the detailed description. We validation our fractional derivatives concept in both objectively and subjectively. The detailed flow of the our data-driven research is described in Figure 4.1.

4.2 Data collection

This section introduces the data collection hardware apparatus, and the details of training dataset and test dataset.

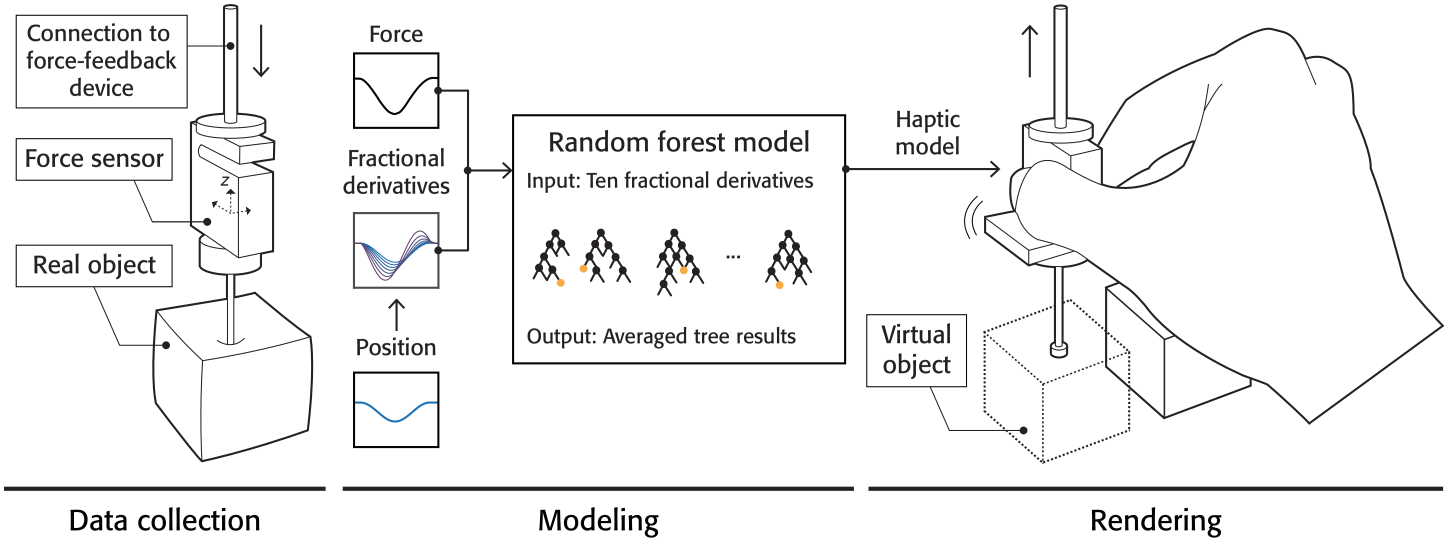
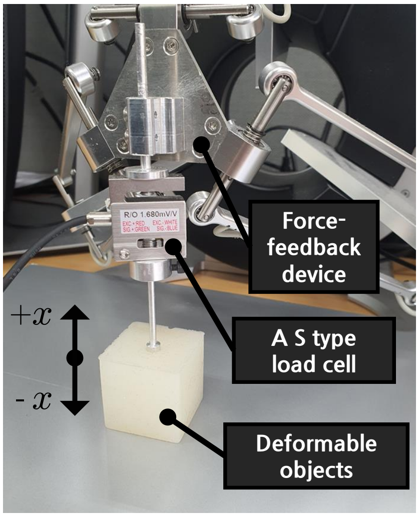


Figure 4.1: While a user haptically interacts with a real deformable object, the user's interaction motion and the object's response force are recorded, as if taking "haptic photographs". The recorded data is processed and used to make a blackbox computational model, e.g., a random forest model, between the motion (input) and the force (output). This data-driven model is used to recreate the interaction dynamics of the real object with a haptic interface in a virtual environment.



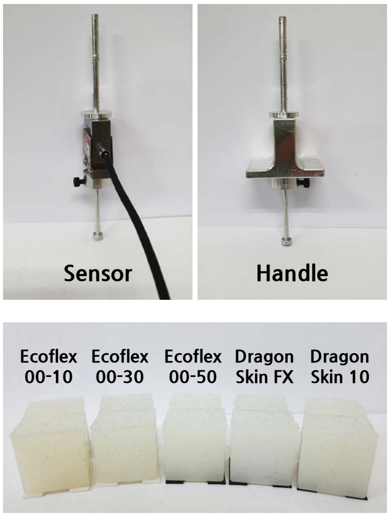


Figure 4.2: A configuration of the data collection. A load cell is connected to the force-feedback device. A deformable object is placed below the load cell.

Figure 4.3: Upper figure shows our data collection setup and rendering setup. Lower figure describes our four materials from Ecoflex 00-30 to Dragon Skin.

4.2.1 Hardware

As shown in Figure 4.2, the experimental setup for data collection consists of a force-feedback device (Omega 3.0, Force Dimension), a load cell (DBCM-2kg, Bongshin), and a deformable object. The software is implemented using an open-source haptic rendering library, CHAI3D. The load cell is attached to the end-effector of the force-feedback device. The other end of the load cell has a tip of 6 mm diameter for interaction with the soft object. The tip is always located at the center of the object's top surface for data collection. The deformation depth is measured by the haptic device, and the force is by the load cell. The data (position and force) is updated at a sampling rate of 1 kHz.

For haptic rendering, the load cell is replaced with an aluminum handle that has a very similar size and weight to the load cell. The upper image in Figure 4.3 compares the sensor and the handle. The gravity compensation constant included in CHAI3D for force rendering is adjusted according to the changed mass of the end effector.

We consider five different silicone materials for data collection. The materials are Ecoflex 00-10, Ecoflex 00-30, Ecoflex 00-50, Dragon Skin FX, and Dragon Skin 10, from the softest to the hardest, all from Smooth-On, Inc. Their detailed information is available in Table 4.1. From each silicon material, a cube with an edge length of 3.5 cm is molded and used in this work; see the lower image in Figure 4.3.

Table 4.1: Silicon materials used in our work.

Material	Shore Hardness	Max. Modeling Depth* (mm)
Ecoflex 00-10	00-10	17.7
Ecoflex 00-30	00-30	14.5
Ecoflex 00-50	00-50	10.4
Dragon Skin FX	2A	7.5
Dragon Skin 10	10A	7.1

* The maximum modeling depth of each material is determined from the stiffness of the object (3.5-cm cube) made from the material and the maximum output force of the force feedback device used (Omega 3.0).

4.2.2 Training Data

Our data collection procedure is automated using the force-feedback device. An inverted cosine signal in (4.1) is commanded as a force control signal to the device's tip:

$$f(t) = -\frac{a}{2}(\cos(2\pi\nu t) - 1), \tag{4.1}$$

where a is the target (peak-to-peak) amplitude in N and ν is the indentation frequency in Hz, respectively. To characterize the object's rate-dependent deformation property for users' general exploratory behavior, we choose three amplitudes, $a \in \{2, 6, 10\}$ (in N), and nine frequencies, $\nu \in \{0.2, 0.25, 0.33, 0.5, 1, 2, 3, 4, 5\}$ (in Hz). For each pair (a, ν), we collect the interaction data for a single indentation cycle. The output force is controlled by position-derivative (PD) control to ensure stable data collection.

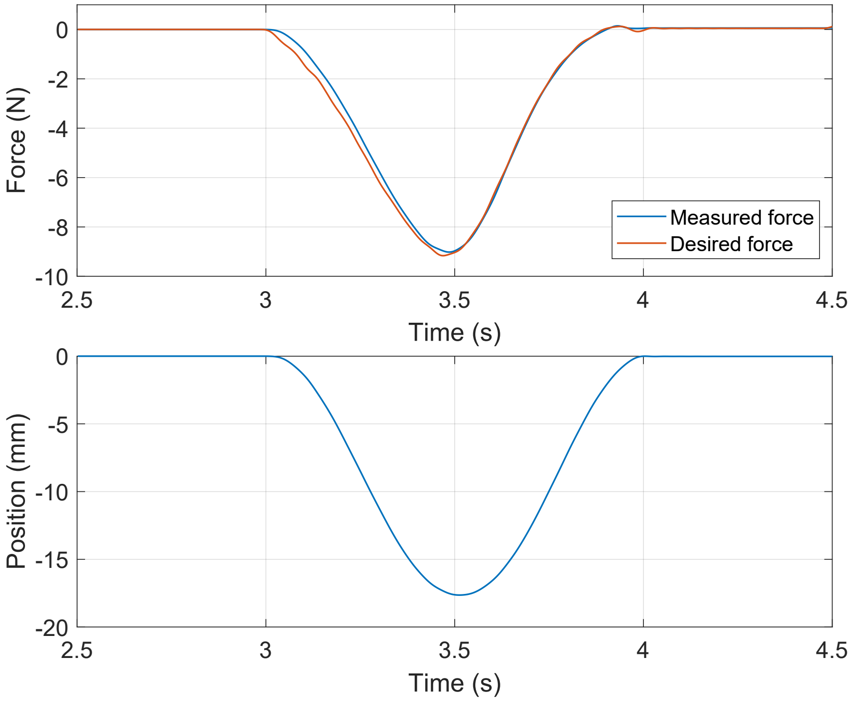


Figure 4.4: Two figures show the 1 Hz of sinusoidal profile. Upper figure shows the measured force from the material and selected points by the sampling method, and lower figures show the measured displacement.

We collect interaction data for each object in Figure 4.3 using the 27 input force profiles (3 amplitudes × 9 frequencies). The total length of the training dataset is 48.85 s. An example of the collected data is provided in Figure 4.4. These datasets are used for training interpolation models (Section 4.3).

4.2.3 Test Data

We collect another dataset for each object to validate the trained interpolation models. The test set must include sufficiently complex and general conditions that can be interpolated from the range of the training data. We generate a complex force signal to indent the object by computing

$$f_i(t) = -\frac{a_i}{2}(\cos(2\pi\nu_i t) - 1), \tag{4.2}$$

for $t \in [0\,\mathrm{s}, 15\,\mathrm{s}]$. a_i and ν_i are randomly chosen between 0.2 N and 1 N and between 0.2 Hz and 5 Hz, respectively, for $i \in \{1, 2, \cdots, 5\}$. The values of $f_i(t)$ after the last zero-crossing before 15 s are zero-padded to make any sum of multiple $f_i(t)$ terms end with zero. Then, we compute

$$f_{sum}(t) = \sum_{i=1}^{5} f_i(t), \tag{4.3}$$

$$f(t) = \frac{10}{\max f_{sum}(t)} f_{sum}(t). \tag{4.4}$$

The five element signals are added in (4.3). The result is normalized in (4.4) by the maximum force amplitude (10 N) used for training data collection. We make ten test input force signals using (4.2)–(4.4) and then collect interaction data for 15 s for each input signal. We apply a zero-phase moving average filter (window size 25) to smooth both the measured position and force data. See Figure 4.6 for examples.

4.3 Modeling

Using the collected interaction data, we learn a nonparametric mapping from the input (position) to the output (force). In this paper, we propose new input features using fractional derivatives, which is free from the rendering instability problem for RF models aforementioned in Section II.

4.3.1 Input Features

The input features are derived from the fractional derivatives of deformation position. Here, we use the sampled sequence of position $x[n]$ and force $f[n]$ from the continuous position $x(t)$ and response force $f(t)$, where n is the discrete time index.

We make an input feature vector as follows:

$$\mathbf{X}[n] = \left(D^{r_1}x[n], D^{r_2}x[n], \cdots, D^{r_{10}}x[n]\right), \tag{4.5}$$

where $D^r x[n]$ denotes the r^{th}-order fractional derivative of $x[n]$ and its order $r_i \in [0,1]$ $(i = 1, 2, \cdots, 10)$. See Chapter III for how we compute $D^r x[n]$. The measured response force sequence is denoted by $f[n]$.

The training dataset of each deformable object includes the 27 position-force signal pairs measured for the 27 input force profiles of different amplitudes and frequencies (Section 4.2.2). We represent them by $x_k[n]$ and $f_k[n]$, respectively ($k = 1, 2, \cdots, 27$). Then, we compute $\mathbf{X}_k[n]$ using (4.5):

$$\mathbf{X}_k[n] = \left(D^{r_1}x_k[n], D^{r_2}x_k[n], \cdots, D^{r_{10}}x_k[n]\right), \tag{4.6}$$

for all $n \in \{1, 2, \cdots, N_k\}$ where N_k is the number of the recorded samples in $x_k[n]$. For each n, $\mathbf{X}_k[n]$ (input feature) is paired with $f_k[n]$ (output force). For compact notation, we stack $\mathbf{X}_k[n]$ and $f_k[n]$ for all n, such that

$$\mathbf{X}_k = \left(\mathbf{X}_k[1], \mathbf{X}_k[2], \cdots, \mathbf{X}_k[N_k]\right), \tag{4.7}$$

$$\mathbf{f}_k = \left(f_k[1], f_k[2], \cdots, f_k[N_k]\right). \tag{4.8}$$

Finally, $\mathbf{X}_k$ and $\mathbf{f}_k$ are stacked over k, i.e., all the 27 training signal sets, such that

$$\mathbf{X}^* = \left(\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{27}\right), \tag{4.9}$$

$$\mathbf{f}^* = \left(\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_{27}\right), \tag{4.10}$$

where $\mathbf{X}^* \in \mathbb{R}^{(10 \times N_k \times 27)}$ and $\mathbf{f}^* \in \mathbb{R}^{(1 \times N_k \times 27)}$. Consequently, $(\mathbf{X}^*, \mathbf{f}^*)$ constitutes the full training dataset.

4.3.2 Model Training

For each object, we estimate an interpolation function Ψ,

$$f[n] = \Psi\left(\mathbf{X}[n]\right), \tag{4.11}$$

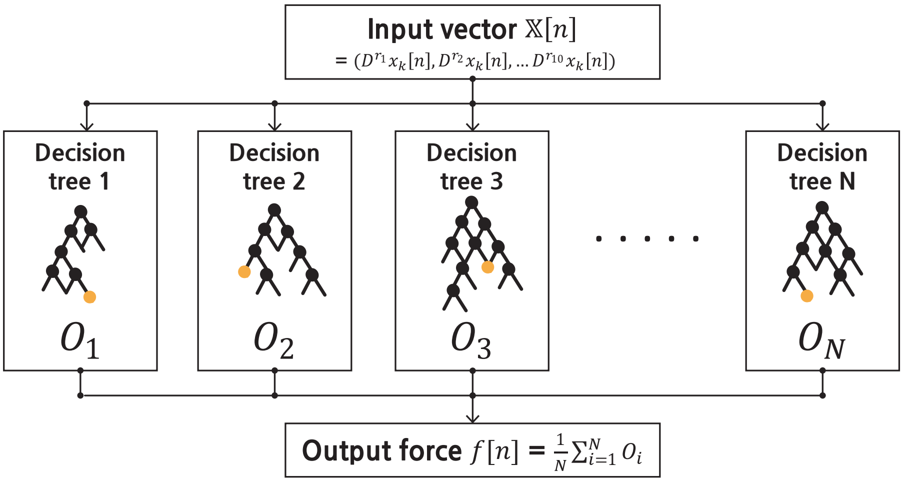

Input vector $\mathbb{X}[n]$
$= (D^{r_1}x_k[n], D^{r_2}x_k[n], \ldots D^{r_{10}}x_k[n])$
Decision tree 1
Decision tree 2
Decision tree 3
Decision tree N
O_1
O_2
O_3
O_N
Output force $f[n] = \frac{1}{N}\sum_{i=1}^{N} O_i$

Figure 4.5: Structure of regression using a random forest.

which predicts the response force $f[n]$ from the FD feature vector $\mathbf{X}[n]$. To this end, we learn a regression forest model on the training dataset prepared for the object. An RF consists of many decision trees and can be used for regression [56]. As illustrated in Figure 4.5, an input vector is fed to many decision trees in parallel, and their output values are averaged to obtain the final output; refer to [56] for more on RFs.

For RF model training, we choose the following parameters: ten orders of FDs in (4.5) $= \{0.05, 0.10, \cdots, 0.50\}$, the number of decision trees = 100, and the stopping criteria = minimum five samples at leaf nodes. We use FDs of many different orders to fully utilize the strength of RFs for automatically searching for meaningful features in a large feature space. Each of the ten orders explicitly represents the different time-dependent behavior (see Figure 3.1 in Chapter III) so that RF training can choose the best rate-related features. For implementation, we use the source codes available in [50]. The resulting interpolation models are denoted by RF-FD models.

For comparison, we also train a radial basis function on input features consisting of FDs:

$$\mathbf{X}[n] = (D^{r_1}x[n], D^{r_2}x[n])\,. \tag{4.12}$$

To choose the best two orders for each object, we train RBF models with all pairs of orders chosen from the ten orders used for the RF-FD models. Then, we select the pair that results in the lowest root mean squared error (RMSE) for the test datasets. The chosen FD orders are shown in Table 4.2, and the resulting models are called RBF-FD models. Note that including more FD terms in the input features significantly increase the computation time for RBF model training. The computational results of such cases are not compared in this paper.

Table 4.2: Fractional derivative orders chosen for RBF models.

Material	r_1	r_2
Ecoflex 00-10	0.30	0.35
Ecoflex 00-30	0.15	0.20
Ecoflex 00-50	0.10	0.15
Dragon Skin FX	0.10	0.15
Dragon Skin 10	0.10	0.15

Lastly, we train another RBF model on the conventional input features of position and velocity:

$$\mathbf{X}[n] = \left(D^0 x[n], D^1 x[n]\right) = (x[n], v[n]) \,. \tag{4.13}$$

The velocity $v[n]$ is estimated using the first-order adaptive windowing method [57]. These models are named RBF-PV, as an implementation of the previous state-of-the-art model [43] for comparison.

All the models are trained on a reduced dataset randomly sampled from the original training dataset, as in [43]. We use only 20% of the training data after repeated tests. The size of the training dataset critically affects the time required for model training. The 20% ratio of the training data to use was determined by empirical tests. Using larger ratio tended to make training of RBF models infeasible.

The two RBF models, RBF-FD and RBF-PV, use the following settings for training: the number of kernel points = 100 and the cubic spline interpolation method.

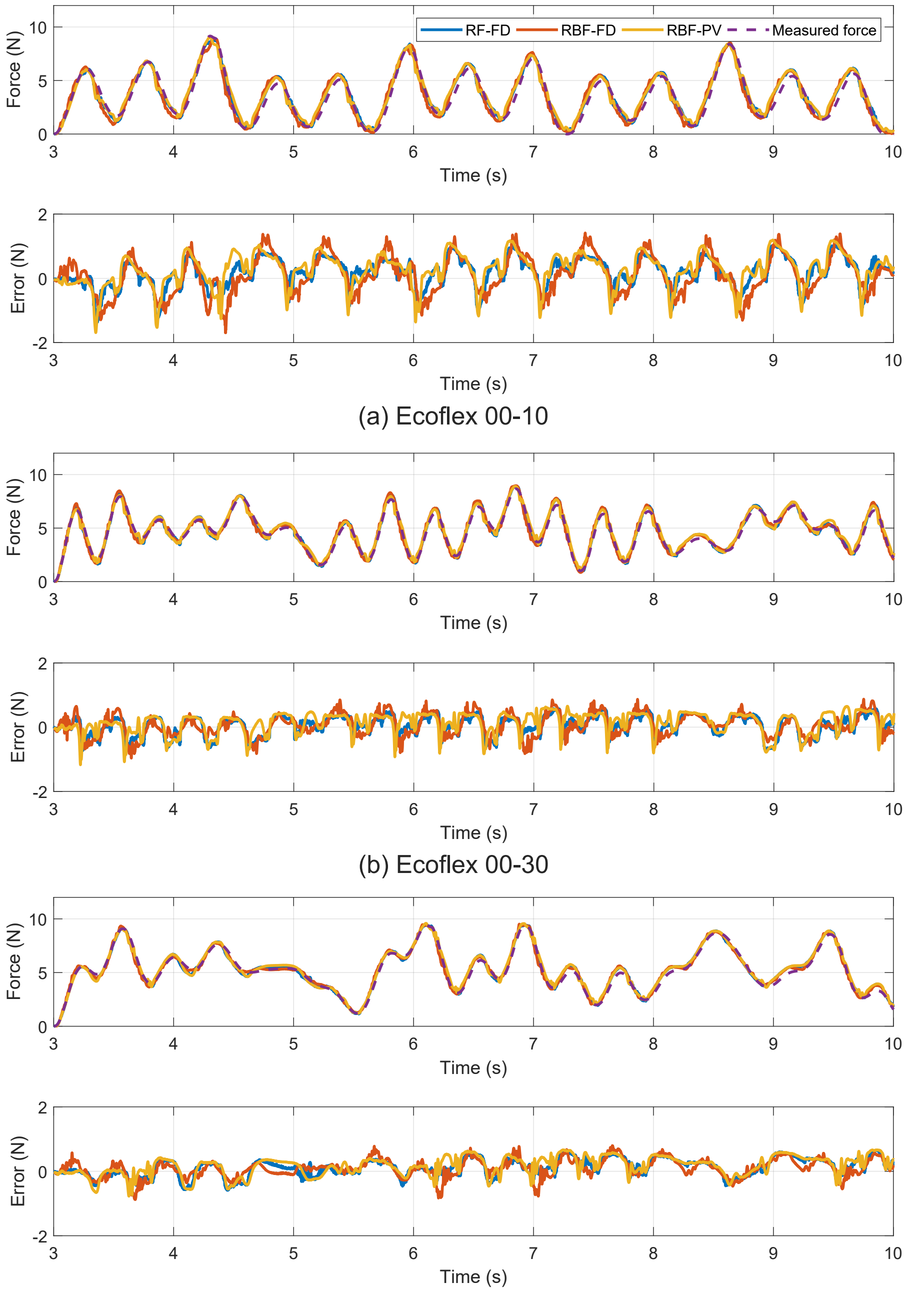
These numbers, as well as the number of trees for RF models, were chosen considering the update rate of haptic rendering of the trained models. As the model size increases, the computation time of rendering becomes longer, which degrades the stability of haptic rendering.

To train the RF-based model on the selected input/output features for each deformable object, we choose the following parameters: ten orders of fractional derivatives= $0.05, 0.10, \cdots, 0.50$, the number of trees= 100, and stopping criteria= minimum five samples at leaf nodes. As the random forests method has the advantages of searching for meaningful features in a large feature vector space, we use different fractional derivatives orders to utilize its strength fully. Each of the orders explicitly represents the different time-dependent features so that the random forests can choose the best time-wise feature.

4.3.3 Modeling Performance

Examples of the force curves predicted by the RF-FD, RBF-FD, and RBF-PV models for all the objects are presented in Figure 4.6 with the measured force curves. They are chosen from the modeling results of the ten test input profiles (Section 4.2.3) to represent the average performance. The lower panels in Figure 4.6 show the errors between the measured and predicted force curves. Generally, the output force profiles closely match the measured force responses for all of the three data-driven models. The absolute prediction errors are less than 0.4 N for all the objects, except Ecoflex 00-10 (the softest material). For Ecoflex 00-10, the absolute errors sometimes increase to around 1 N.

We consider the absolute percentage error (APE) as the difference between the measured and predicted forces divided by the measured force. For perceptual similarity, the APE should lie below the just noticeable difference (JND) of human force perception. Figure 4.8 presents the APE vs. the measured force for each object's trained models. Each plot also shows the JND curve of force perception for comparison. The

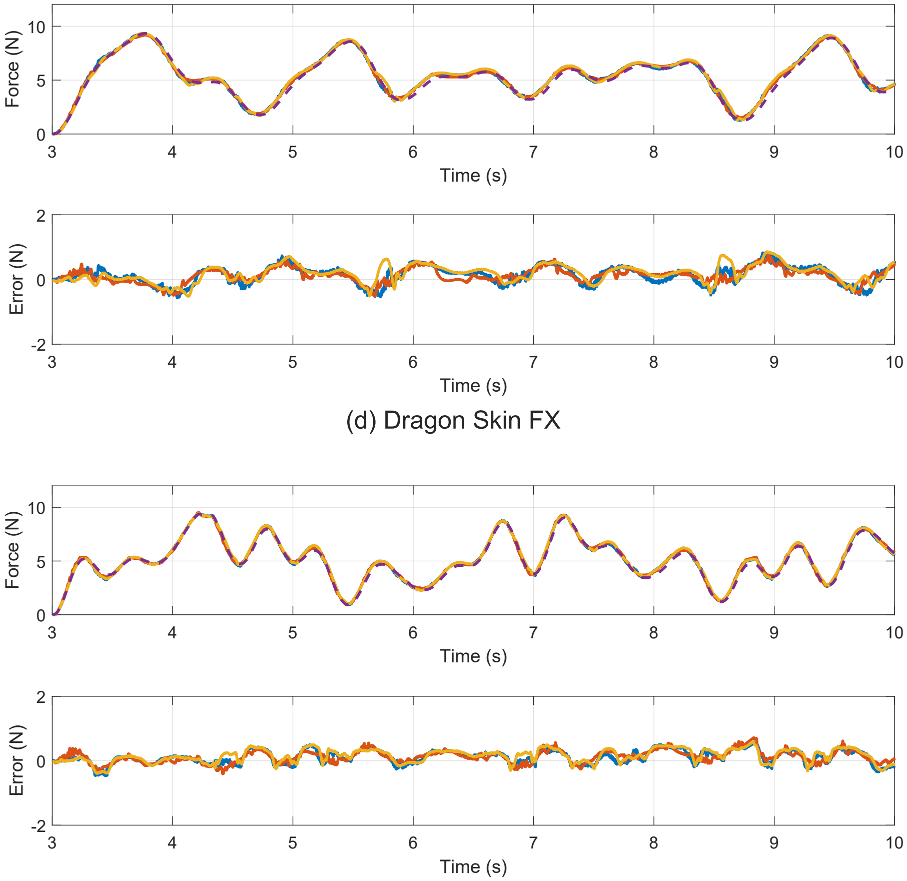


(a) Ecoflex 00-10

(b) Ecoflex 00-30

(c) Ecoflex 00-50

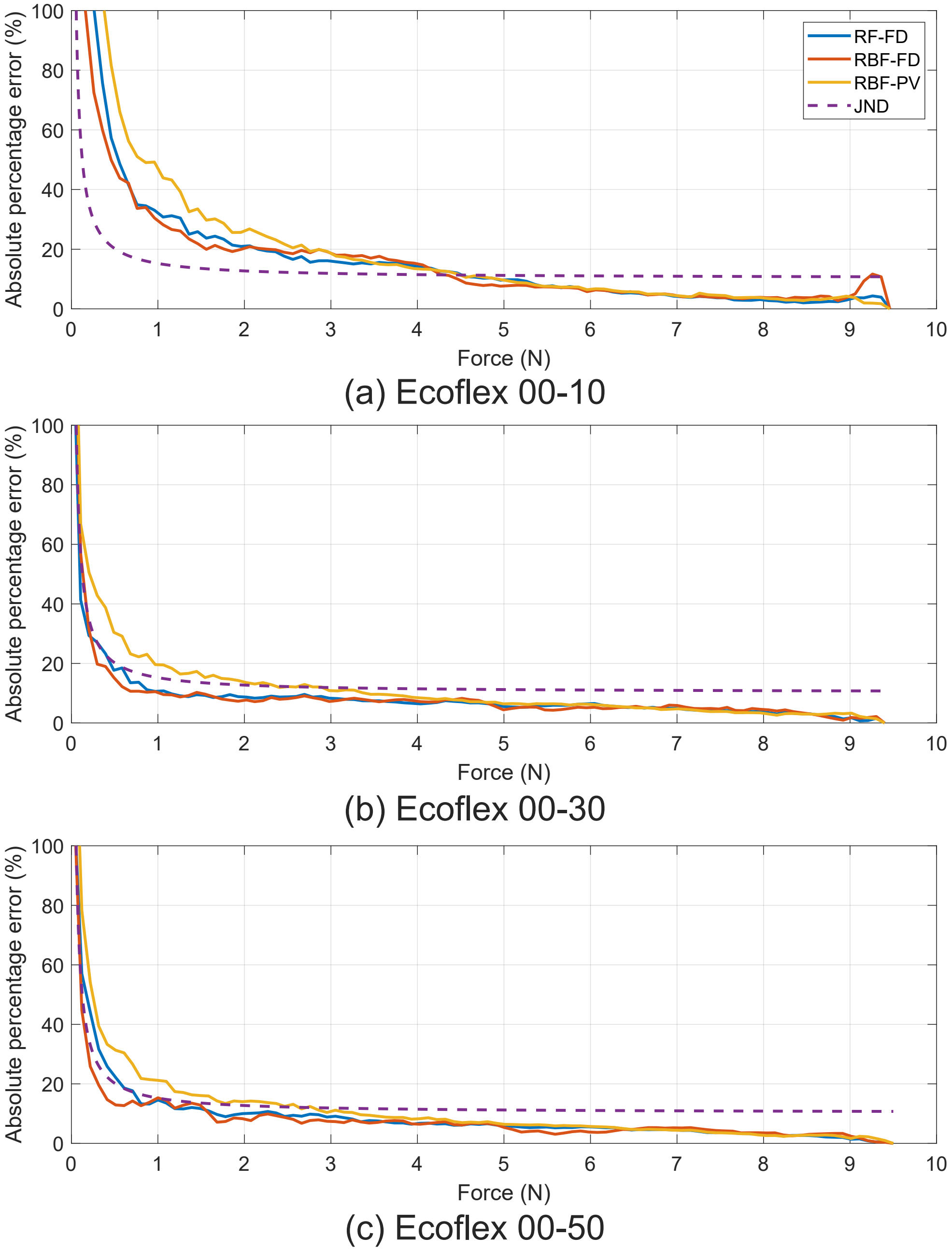
Figure 4.6: Exemplar modeling results on test signals.



(d) Dragon Skin FX

(e) Dragon Skin 10

Figure 4.7: Exemplar modeling results on test signals.

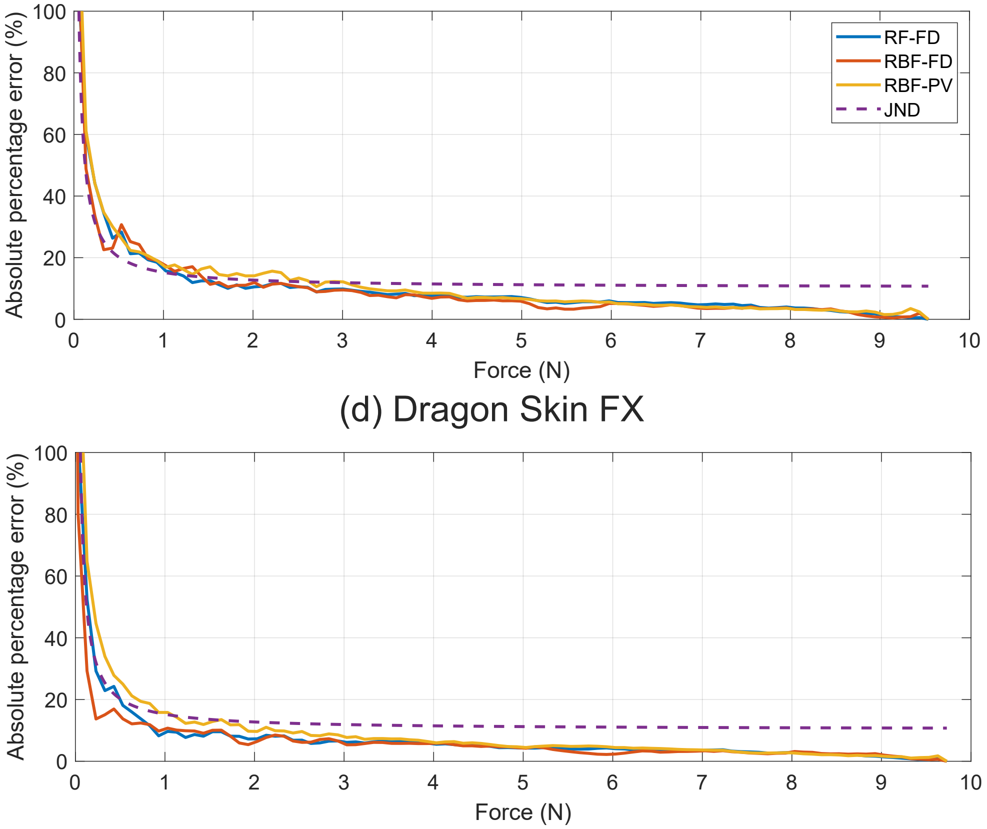


(a) Ecoflex 00-10

(b) Ecoflex 00-30

(c) Ecoflex 00-50

Figure 4.8: Absolute percentage errors on the test dataset.



(d) Dragon Skin FX

(e) Dragon Skin 10

Figure 4.9: Absolute percentage errors on the test dataset.

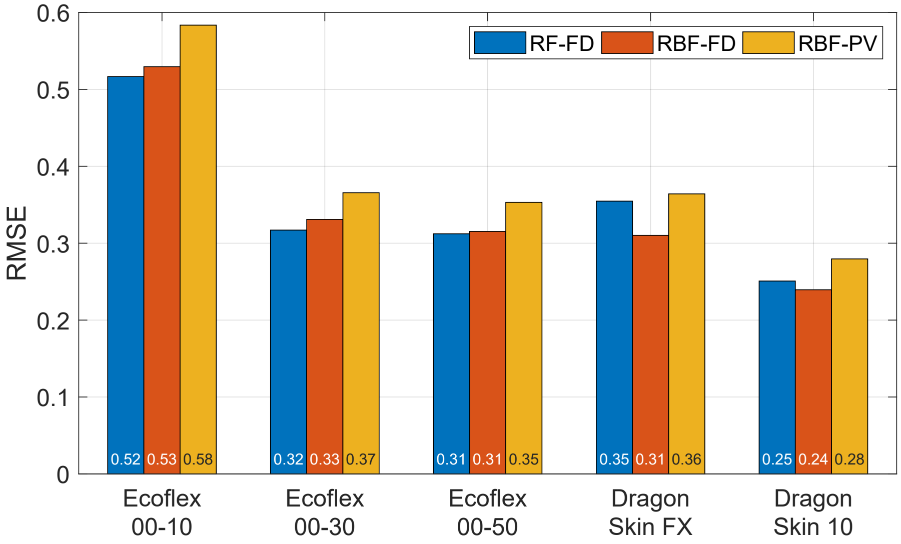


Figure 4.10: RMSEs of the three interpolation models.

JND curve is initially very large and then rapidly decreases as the reference force increases, finally plateauing to the constant Weber fraction (10%) [58]. The APEs for all the models lie below the JND curve for the reference force larger than 2 N for all the objects, which is generally acceptable [59, 48]. Among the three models, the two FD methods show lower APEs than RBF-PV.

For each object, we also compute the RMSEs averaged over the ten test datasets. The results are shown in Figure 4.10. RF-FD and RBF-FD, trained on the fractional derivatives, result in slightly lower RMSEs than RBF-PV trained on the standard position and velocity data. RF-FD and RBF-FD have very similar RMSEs for all the objects.

Last, we report the modeling computation time for both feature computation and model training. The measurement was performed on a regular PC (Windows 10, i7-11700k CPU, 32 GB memory) using Matlab. Results are shown in Figure 4.11. The computation times were similar regardless of the real object for all three models. Their means were 11.9 s, 126.7 s, and 126.7 s for RF-FD, RBF-FD, and RBF-PV, respec-

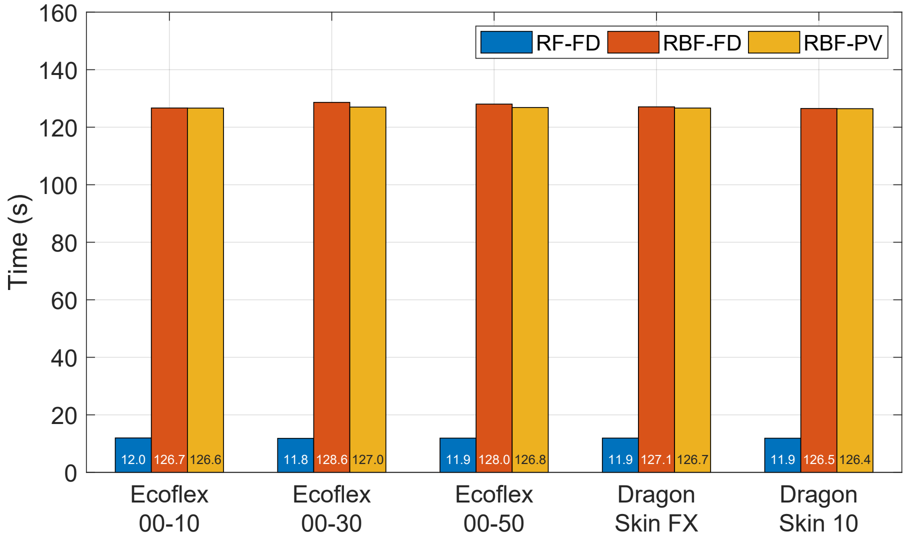


Figure 4.11: Average modeling times of the three interpolation models.

tively. Therefore, the RF-FD method requires less computation by more than ten times than the two RBF methods. The actual modeling time will decrease in proportion to the number of surface contact points when we extend our modeling approach to inhomogeneous objects as in [46].

The above results indicate that RF-FD provides similar accuracy to the two RBF methods at greatly superior modeling cost. Note that the prediction accuracy and computation time can be further improved if the parameters are optimized per object. For example, Figure 4.12 shows the variable importance of the ten FD features obtained from the RF training for the five real objects. A variable importance is a relative indicator of how much information the corresponding feature provides to a random forest. The plots show that fractional derivatives of low orders are dominant for hard objects (low viscosity), whereas those of medium orders (0.3-0.5) are more important for soft objects (high viscosity). Including higher order terms close to 1.0 is not necessary for the objects tested, although it may be for highly viscous objects. We may further improve the prediction and/or computation performance by using only the features of

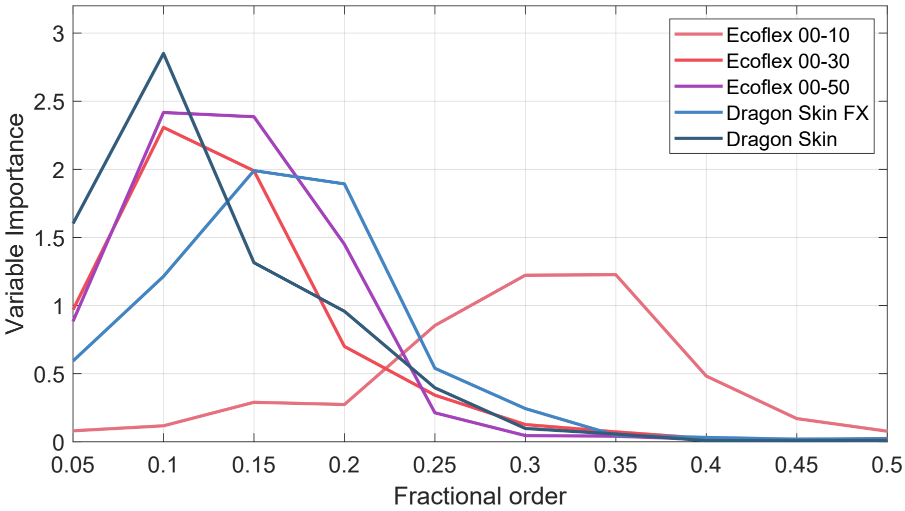


Figure 4.12: Variable importance from RF training results.

high importance for each object.

4.4 Haptic Rendering

For haptic rendering, we recreate the deformation dynamics of a real object by simulating its data-driven model and then generating the response force. To this end, the end-effector of the haptic device is replaced by the handle shown in Figure 4.3.

Our haptic rendering program is implemented under Microsoft Windows using C++ and CHAI3D, similar to the data collection program. Since we train RF and RBF models using MATLAB, the resulting models are loaded into the haptic rendering program. A loader file for RF models includes all information required to construct and simulate a regression forest. Given an input feature vector, all decision trees in the RF are traversed, and their results at the leaf nodes are averaged to determine the final output force. A loader file for RBF models delivers information about the centers of radial basis functions and their weights. Using these values, our program computes the rendering force for an input feature vector of RBF models.

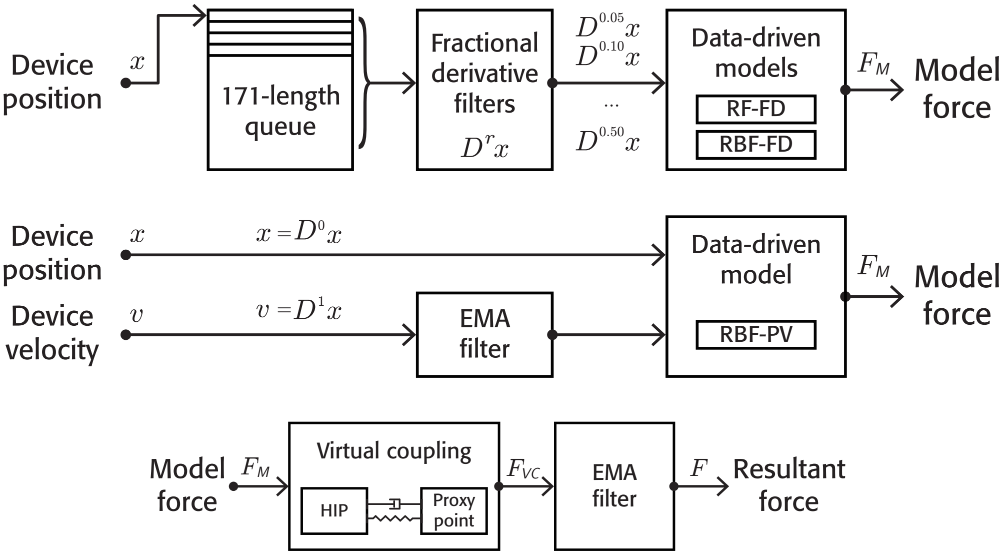


Figure 4.13: Procedure of haptic rendering. Data-driven model simulation for RF-FD and RBF-FD (a) and for RBF-PV (b). Virtual coupling to improve rendering stability (c).

The overall procedure of haptic rendering is depicted in Figure 4.13. The position of the haptic tool is denoted by x. As shown in Figure 4.4, the past position samples for 171 ms are queued in a vector. It is convolved with a fractional derivative filter of the same length to compute fractional derivative features for RF-FD and RBF-FD models. The FD filter can be precomputed because our models use the fixed fractional orders (0.05, 0.1, $\cdots$, 0.5), fixed time difference (1 ms), and limited length (171); see Chapter III. The FD features are input to a data-driven model (RF-FD or RBF-FD) to compute the virtual model force f_M. For RBF-PV computation, shown in Figure 4.4, we use the tool velocity v provided by the haptic device's driver. Its output is smoothed using an exponential moving average (EMA) filter with the weighting factor α for the current term of 0.7. These position and velocity are the input to an RBF-PV model, which calculates the output force f_M.

The data-driven model output f_M is used as the input to a virtual coupling algo-

Algorithm 1 Virtual Coupling Algorithm

```
function VIRTUAL_COUPLING($x, v, f_M$)
    if $x_{proxy} = NULL$ then
        $x_{proxy} \leftarrow x$
        $v_{proxy} \leftarrow v$
        $f_{VC} \leftarrow 0$
    else
        $f_{VC} \leftarrow -k(x_{proxy} - x) - b(v_{proxy} - v)$
        $f_{proxy} \leftarrow f_M + f_{VC}$
        $a_{proxy} \leftarrow f_{proxy}/m$
        $v_{proxy} \leftarrow v_{proxy} + a_{proxy}\Delta t$
        $x_{proxy} \leftarrow x_{proxy} + v_{proxy}\Delta t$
    end if
    return $f_{VC}$
end function
```

rithm to improve rendering stability, as depicted in Figure 4.4. Its computational steps are specified in Algorithm 1[1]. In the pseudocode, x_{proxy}, v_{proxy}, and a_{proxy} denote the virtual proxy position, velocity, and acceleration, respectively. x_{proxy} is initialized with $NULL$ before the initial contact. f_{proxy} is the force applied to the virtual proxy to update its movement, and f_{VC} is the force output of the virtual coupling. Δt is the sampling time. The algorithm simulates the simple dynamics of a virtual spring-damper system connecting the device position and the virtual proxy. The virtual proxy is treated as a quasi-static point mass, and its movement is simulated accordingly. The algorithm returns the coupling force between the two points. We use the following parameters: spring constant $k = 1.5$ N/mm, damping coefficient $b = 20$ Ns/mm, and mass $m = 50$ g, which are manually tuned. The output force f_{VC} of virtual coupling is smoothed further by an EMA filter with $\alpha = 0.7$. This final output force f is sent to the force-feedback device for haptic rendering. The effects of virtual coupling and

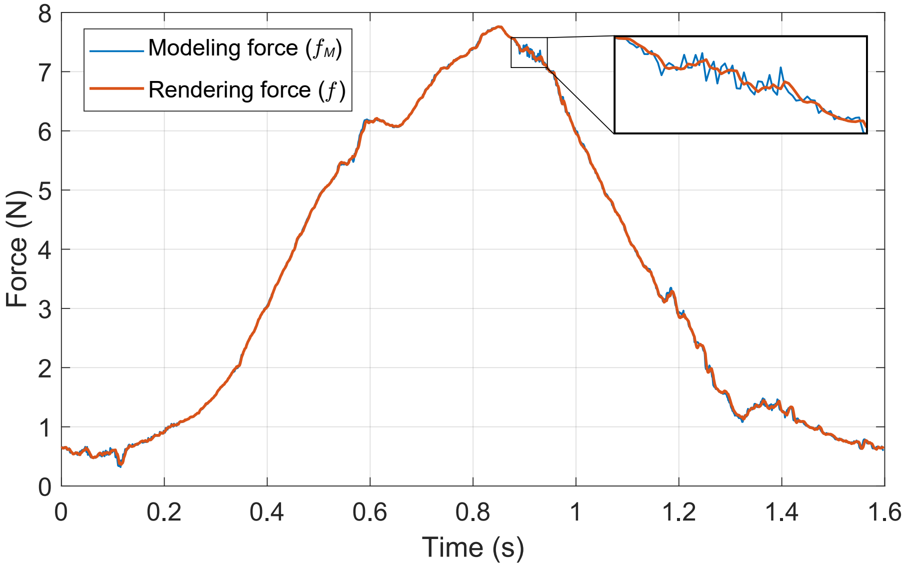


Figure 4.14: Comparison between the force computed from data-driven models and its smoothed force by virtual coupling and EMA filtering.

EMA filtering are illustrated in Figure 4.14 using an example.

The above procedure is executed repeatedly in a dedicated thread. Table 4.3 shows the three data-driven models' update rates measured using a regular PC (Windows 11, i7-11700k CPU, 32 GB memory). The update rates are similar regardless of the real object used for modeling. RBF-PV shows the fastest update rate (mean 1982.6 Hz), followed by RBF-FD (mean 990.3 Hz). This difference is due to the use of FD features. RF-FD shows the lowest update rate (mean 788.3 Hz), indicating that rendering force computation using RF models using 10 FD features is slower than RBF models using 2 FD features by approximately 20%. Nonetheless, all of these update rates are sufficient for the haptic rendering of relatively soft deformable objects.

When computing FD features, we use the FD coefficients precomputed assuming 1-ms sampling time. The average update times for RF-FD and RBF-FD are 1.27 ms

[1] We thank Arsen Abdulali and Seokhee Jeon, the authors of [48], for providing the working codes of virtual coupling for deformable objects.

Table 4.3: Update rates (Hz) of the three data-driven models.

Material	RF-FD	RBF-FD	RBF-PV
Ecoflex 00-10	787.5	993.4	1984.7
Ecoflex 00-30	794.7	988.6	1980.1
Ecoflex 00-50	792.7	990.3	1983.4
Dragon Skin FX	781.4	990.3	1982.1
Dragon Skin 10	785.0	988.9	1982.9
Mean	788.3	990.3	1982.6

and 1.01 ms (Table 4.3). Thus, we need to examine whether the force output errors caused by the sampling time difference in RF-FD are significant. RF-FD has a sampling time of 1.27 ms, and the derivative orders that have high variable importance are lower than 0.4 (Figure 4.12). The errors in the value of the fractional derivative are negligible, according to Figure 3.2 in Chapter III. Thus, the slightly lower sampling time of RF-FD is expected not to cause practically noticeable problems. For confirmation, we resample the actual rendering results of position and force to have the exact 1-ms sampling time by linear interpolation and compare them with the rendering results. An example is shown in Figure 4.15 for the RF-FD model of Ecoflex 00-50. The actual rendering results with approximately a 1.27 ms sampling time and the resampled results with the exact 1-ms sampling time show very similar time-force curves. This close similarity is also observed for the other objects. Therefore, the RF-FD model is quite robust to the variability in the update rate of haptic rendering, at least in the range of real materials tested in this work.

4.5 Perceptual Experiment

We performed a user study to validate the perceptual performance of our data-driven haptic modeling and rendering framework for viscoelastic deformable objects. Participants' task was to compare the similarity between real and virtual objects and

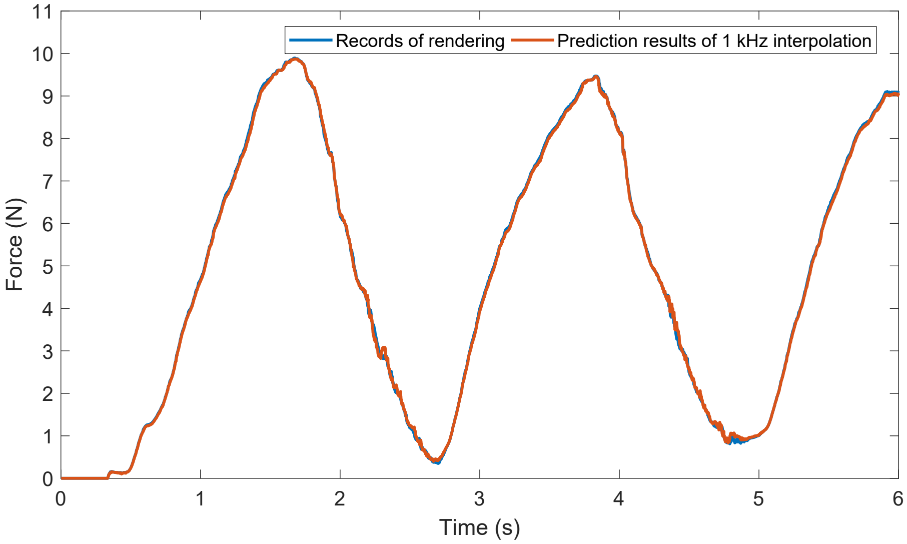


Figure 4.15: Time-force curves for the actual rendering results of the RF-FD model for Ecoflex 00-50 and the resampled results at the sampling time of 1 ms.

then represent it using a number. This similarity rating paradigm has been frequently used in related studies to data-driven haptics [46, 37, 60, 61, 39, 48].

4.5.1 Methods

Participants

We recruited 18 adults (5 females and 13 males, average age 24.2 years) for this experiment. None of them had a history of neurophysiological disorders. They were provided with written and verbal instructions about the experiment. They were paid KRW 15,000 ($\simeq$ USD 12) for compensation.

Task

Participants sat comfortably in front of a computer monitor. They controlled the force-feedback device (Omega.3) with their right hands and used their left hands to enter responses using a keyboard; see Figure 4.16. A physical wall was placed between

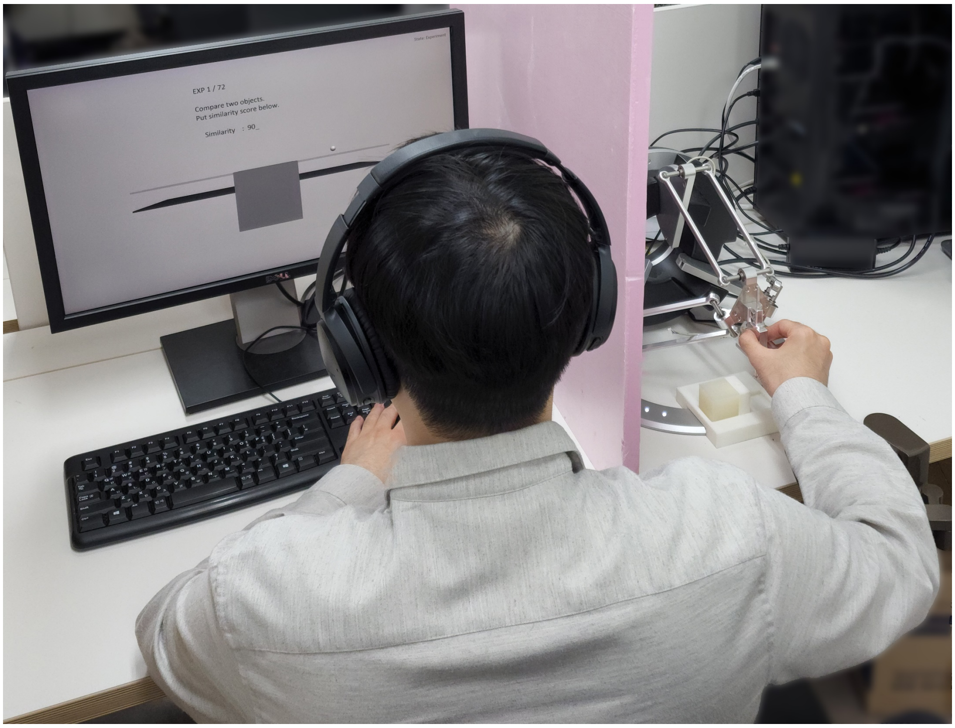


Figure 4.16: Setup of the perceptual experiment. A participant controls the force-feedback device while watching the monitor screen. The force-feedback device and deformable objects are visually blocked by the physical wall.

the haptic device and the monitor screen to block any visual cues regarding the device movements and the real/virtual objects.

Figure 4.17 illustrates the experimental setup. When a real object was placed on the left side, the corresponding virtual object was rendered on the right side, and vice versa. Participants explored both objects while pressing and releasing them in the vertical direction using the haptic device without a time limit. To guide the haptic exploration, the monitor screen displayed the haptic interface point (HIP), a visual guideline, and a contact line. The HIP represents the position of the device tool tip in the virtual environment. The contact line was aligned with the top faces of the real and virtual objects. The visual guideline indicated the maximum permissible penetration

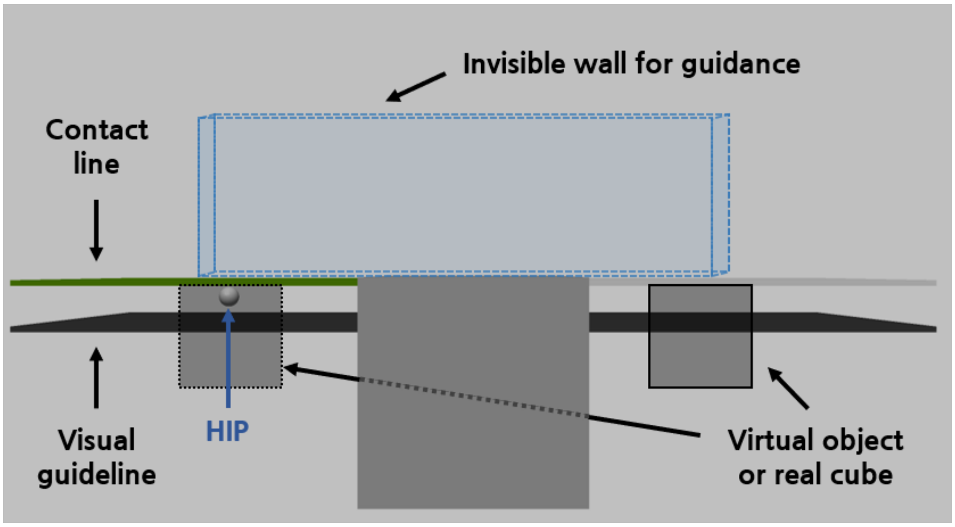


Figure 4.17: Configuration of real and virtual objects used in the perceptual experiment. Note that on the monitor screen, real or visual objects are not displayed.

depth (7.5 to 11 mm) used for training the models. An invisible virtual wall was also rendered to define the workspace limit while helping participants interact around the centers of the real/virtual object's top faces.

We provided two specific instructions to participants to ensure the rendering to take place in the range covered by the data-driven models. First, they were asked not to penetrate the objects exceeding the visual guideline. Otherwise, a warning message was given. This precaution was to keep the rendering force stay within the force range of the training data. Second, they were asked not to move too fast while touching the objects. An error message was shown when their movement velocity was four times larger than the maximum velocity used to generate the training data. Additionally, participants wore noise-canceling headphones that played white noise to prevent any possible effect of external sound on the experimental results.

Conditions

We tested 21 experimental conditions: the 5 deformable objects (Ecoflex 00-10, Ecoflex 00-30, Ecoflex 00-50, Dragon Skin FX, and Dragon Skin 10) × the 3 trained models (RF-FD, RBF-FD, and RBF-PV) + 5 upper bound conditions (per material) + 1 lower bound condition. The upper bound conditions were to compare two identical real objects included to obtain maximum similarity scores. In the lower bound condition, participants compared the two most different real objects: Ecoflex 00-10 (the softest) and Dragon Skin 10 (the hardest). This condition provided a perceptual anchor for the most dissimilar case.

The haptic rendering functions for RF-FD and RBF-FD were executed as fast as possible at the update rates shown in Table 4.3. For RBF-PV, the update rate was fixed at 1000 Hz.

Procedure

The experiment consisted of three sessions of identical design. In one session, each experimental condition was tested once, except the low bound condition. The low-bound condition was repeated four times. Thus, one session included 24 trials. The order of the experimental conditions was randomized per session and participant. Each participant completed all of the three sessions in a within-subject design.

In each trial, participants compared two deformable (real or virtual) objects. The experimenter manually switched the real objects under the tip of the haptic device. The haptic device generated no force while a real object was explored. The positions (left or right) of the real and virtual objects were randomly selected per trial. Participants compared the haptic responses of the two objects sufficiently without a time limit. Then, they rated the perceptual similarity between the two objects by entering a number between 0 and 100 using a keyboard.

Before performing the main sessions, participants had a training session. The training session had the same procedure as that of the main session. Participants familiarized themselves with the haptic device and the experimental task. The experimenter provided feedback on participants' behavior of executing the task whenever necessary.

At the end of the experiment, participants had a debriefing session and answered a questionnaire consisting of the following five questions:

Q1 Was there any unstable or abnormal behavior from the haptic device during the experiment?

Q2 What criteria have you opted for rating similarity?

Q3 Could you distinguish between real and virtual objects?

Q4 If your answer to Q3 is yes, then what were your criteria?

Q5 Which factors/aspects can improve the rendering?

In addition, we recorded the trajectory of the HIP in all trials and sessions.

4.5.2 Results

Outlier removal

Some participants showed inconsistent responses in the experiment. For objective assessment, we computed the standard deviations (SDs) of the similarity scores collected in the same experimental conditions for each participant. Two participants, P4 and P13, showed very high inconsistency. Their SDs averaged across the 21 experimental conditions exceeded 30, while the mean SDs of the other participants were between 3.87 and 24.33. We also checked possible inconsistency due to the stimulus position by averaging the similarity scores collected when the trained model was placed left or right. Two participants, P6 and P13, failed this test. Their left-right difference scores were over 30, while the average of the other participants was 5.47. As a result, we removed the experimental data of the three participants, P4, P6, and P13, and proceeded with the data of the other 15 participants.

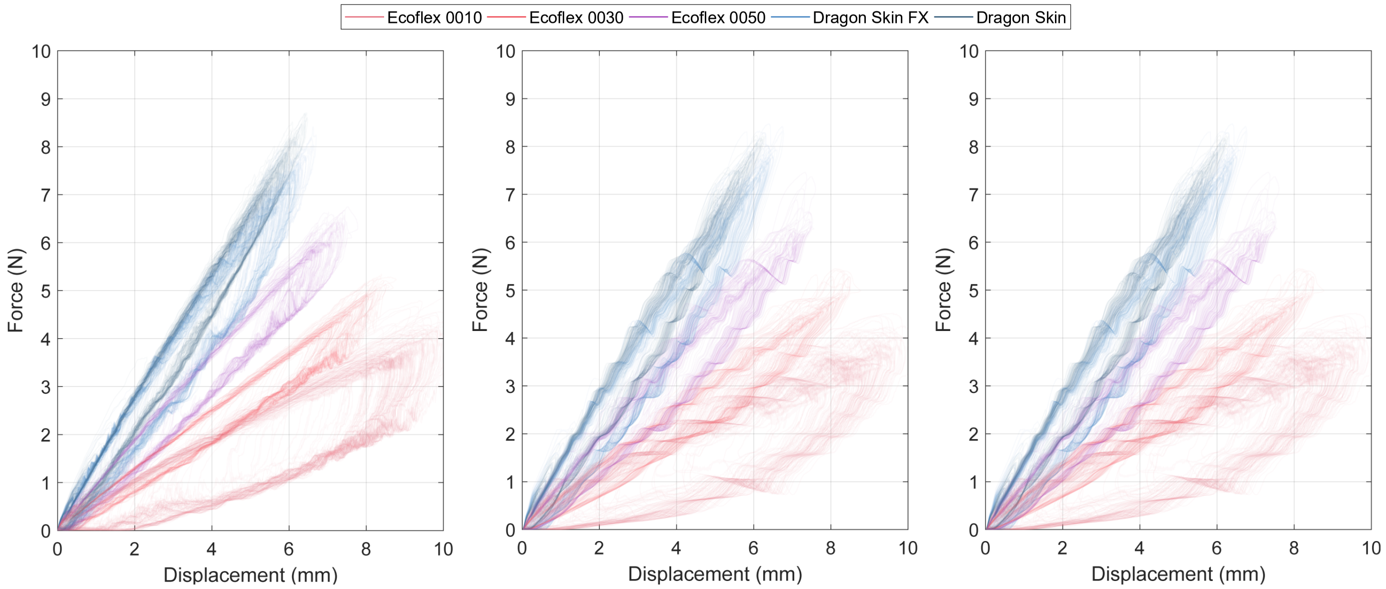


Figure 4.18: Displacement-force hysteresis curves constructed from the HIP trajectories of all participants.

HIP Trajectories

Figure 4.18 shows the displacement-force hysteresis curves combining all the HIP trajectories and the command forces, collected from the interaction data of one participant, for each virtual object and data-driven model. A hysteresis curve for a typical viscoelastic object exhibits different behaviors when the object is pushed (loading) or released (unloading). In Figure 4.18, each curve's upper and lower parts represent the loading and unloading behaviors, forming an elliptical curve. Harder objects, e.g., Dragon Skin 10, shows a smaller hysteresis curve with higher stiffness, while softer objects, e.g., Ecoflex 00-10, shows a larger hysteresis curve with low stiffness. These behaviors are consistent with those reported in the literature for real and virtual viscoelastic objects [62, 59, 43]. Other participants resulted in similar displacement-force curves.

Similarity Scores

Figure 4.19 presents the collected similarity scores SS in box plots for all the 21 experimental conditions. In the upper-bound conditions (comparison between the same real objects), the mean SS ranged from 81.6 to 88.5, with a grand mean of 84.2. In the lower-bound condition (comparing the softest and hardest real objects), the mean SS was very low at 8.7. The mean SS values of the three data-driven models were substantially greater than the lower bound, and they were comparable to the upper bounds. For the five materials, the RF-FD models resulted in the mean SS values between 73.8 and 85.5, RBF-FD between 72.0 and 81.7, and RBF-PV between 77.9 and 82.7.

For statistical analysis, we applied two-way repeated-measures ANOVA with the independent variables of $Method$ (RF-FD, RBF-FD, RBF-PV, or REAL) and $Material$ (Ecoflex 00-10, Ecoflex 00-30, Ecoflex 00-50, Dragon Skin FX, and Dragon Skin 10) to SS. The data of the lower-bound condition were not included. $Method$ ($F(3, 45) = 6.74, p < 0.001$) and the interaction between $Method$ and $Material$

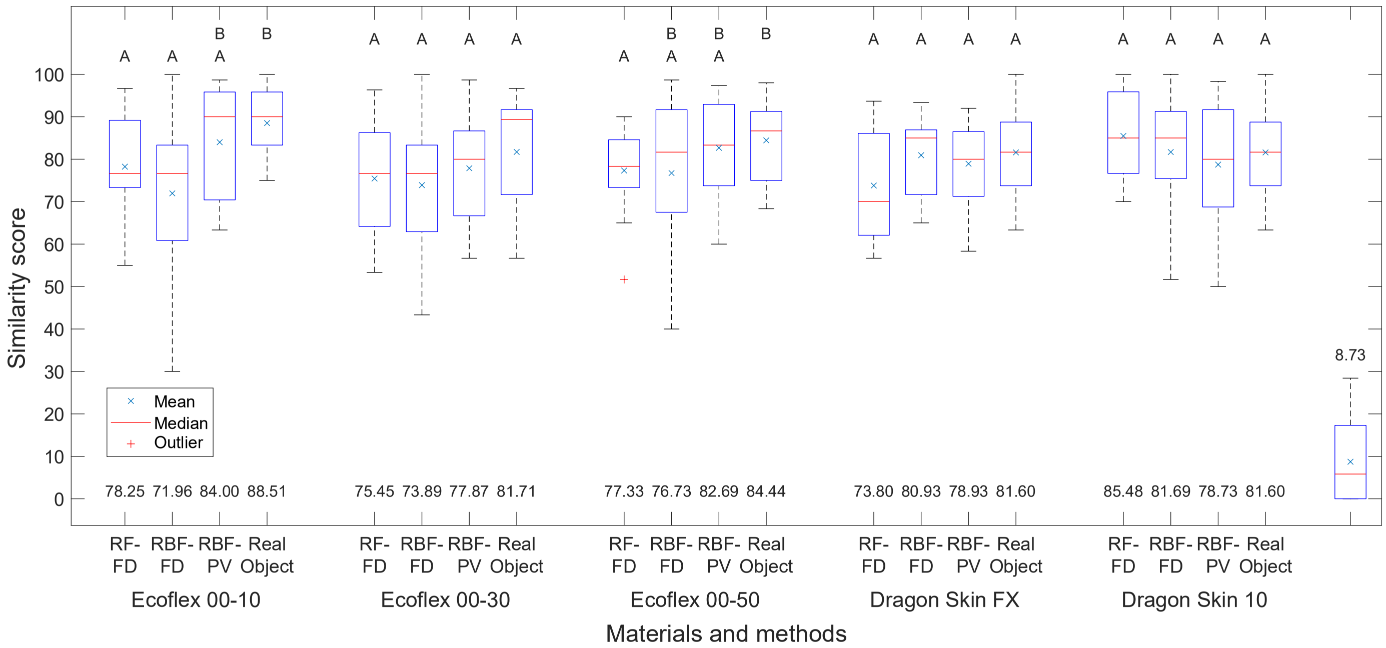


Figure 4.19: Similarity rating results of the perceptual experiment. The means of each condition are specified right above the horizontal axis. For each material, conditions without statistically significant differences are grouped using the same letters.

($F(12, 180) = 2.84, p = 0.001$) had significant effects on SS, but $Material$ did not ($F(4, 60) = 2.08, p = 0.094$).

To further examine the significant interaction term, we conducted simple effect tests on the SS of each $Material$. $Method$ had a significant effect at Ecoflex 00-10 ($F(3, 42) = 7.30, p < 0.001$), Ecoflex 00-50 ($F(3, 42) = 3.40, p = 0.026$), and Dragon Skin FX ($F(3, 42) = 2.87, p = 0.047$). We ran Tukey's post-hoc tests for the significant cases. For Ecoflex 00-10, a significant difference in SS existed between the two FD models and the upper bound condition. For Ecoflex 00-50, RF-FD and the upper bound condition showed a significant difference. For Dragon Skin FX, no significant differences were found among the four conditions of $Method$.

Questionnaire

The participants' responses to the five questions are summarized as follows.

Q1 Eleven out of the 15 participants sometimes felt (unstable) vibrations while exploring the virtual objects.

Q2 All of the participants regarded the response force as the main criteria for similarity rating. Three participants also considered the change in response force when they varied the haptic exploration velocity.

Q3 Eight participants could distinguish virtual objects from real objects.

Q4 The major sensory cue for detecting virtual objects was oscillations that the participants sometimes experienced at deep penetration.

Q5 Ten participants provided suggestions to improve the haptic rendering. Six of them were about removing unrealistic oscillations at deep penetration.

4.5.3 Discussion

The similarity scores were bounded below by 8.7 (between the most different real objects) and above by 84.2 (between the same real objects). This resulted from the participants' general tendency to avoid extreme responses. All of the data-driven models led to high similarity scores comparable to or slightly smaller than the upper bound. The mean similarity scores of RF-FD, RBF-FD, and RBF-PV across all the objects were 78.1, 77.0, and 80.4, respectively. No significantly different cases were found among the pairs of the three data-driven models for any of the five materials (Figure 4.19). Therefore, all of the data-driven models and rendering methods are highly and similarly effective in replicating the deformation dynamics of a real object.

In one of the related studies [63], similar perceptual similarity scores were obtained for the same materials using the FEM-based modeling approach. The similarity scores of our RF-FD model were 75.4 and 77.3 for Ecoflex 00-30 and Ecoflex 00-50, respectively (Section 4.5.2). The FEM-based approach in [63] provided the similarity scores of 82.7 and 84.9 for the same two material cases. The scores of both studies are very high, although the absolute scores cannot be compared in a statistically meaningful way due to the difference in the detailed methods and environment in the experimental setups. Therefore, our work demonstrates that viscoelasticity can be modeled on a smaller amount of data in a greatly shorter time than any other models from the literature, while preserving the high accuracy.

The unrealistic oscillation problem mentioned by many participants generally occurred when the user touched a virtual object with a very fast velocity or penetrated very deep into the object. In our data-driven models, the dynamics information for such high velocity or deformation is not defined, and the rendering is likely to exhibit unexpected behaviors. In fact, this problem is common to all data-driven methods. One remedy can be extrapolation with a well-defined physics-based model if the input variables exceed the data collection ranges.

V. Perceptual sampling method

5.1 Sampling methods

Our next step is improving our modeling and rendering method in Section 4.3. It includes reducing the number of input data, reducing the modeling error rate, or improving the perceptual rendering quality. In this chapter, we define and apply a perceptual sampling method to effectively select the meaningful points from the training set. Then, we compare the modeling and rendering results with the previous results.

5.1.1 Review on random sampling method

In Section 4.3, we randomly sampled 20% from the collected data, which was selected considering the stability in rendering. In the pre-training, we trained randomly-sampled models from 1% to 30% and tested them on the test set. The detailed RMSE results on the percentages of the training samples are in Figure 5.1. The RMSE values are calculated upon the ten test set results. In this report, we trained ten repetitions of models and averaged their results to minimize the random effect. The solid lines are the results of Ecoflex 00-10, the dashed lines are Ecoflex 00-30, and the dotted lines are Dragon Skin 10. The random forest models, shown in blue lines, show that RMSE values were almost constant regardless of the percentage, while the radial basis function models, RBF-FD and RBF-PV, which are red and yellow lines, respectively, show that the RMSE values were saturated at the 20%. Our empirical test also found that 20% of training samples ensure a stable rendering compared to the smaller number of samples.

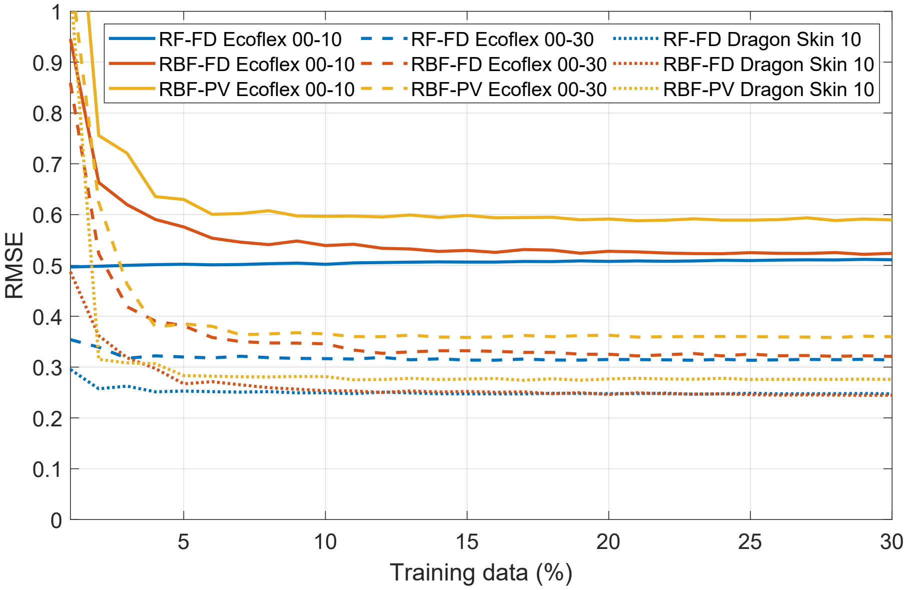


Figure 5.1: Training percentage versus RMSE. The three different colors shows the training models and the three different line style

5.1.2 Perceptual sampling method

We observed room for improvement throughout choosing the random sampling percentage. First, the sampled points are imbalanced among the length of the signal, which is inversely proportional to the frequency. Assume that we choose 10% of the data in the random sampling method. Then, the number of the selected points from the 0.2 Hz cycle signal is 500, while the number from the 5 Hz cycle signal is 20. This length-based selection includes fewer data from the fast movement case. Second, furthermore, the meaningful point such as the highest amplitude point is sometimes omitted. Lastly, there are many duplicated points and they can be affected on the modeling time. To this end, we define a perceptual sampling method which is based on Weber's law. This idea is inspired by the sampling method for the haptic telecommunication problem [51].

We employ the same non-uniform adaptive sampling approach/algorithm to reduce the recorded dataset. The approach is applied on each sample of the force curve $\mathbf{f}_k$. The algorithm starts with the first sample (i.e., $f_k[1]$)) and considers it as a reference force f_r. Next, we progressively search for the sample index j for which

$$\frac{|f_k[j] - f_r|}{0.1 + |f_r|} > \delta$$

is satisfied. The selected sample $f_k[j]$ is considered to have significant information with respect to the reference sample f_r. The sample index j is stored, and the reference force f_r equals $f_k[j]$ for the next search. We repeat this process on each force curve of k^{th} signal of the dataset separately and store all the indices of the selected force samples. This approach may discard local maxima points in the force curves, which are very much essential for training the modeling algorithm. To accommodate this, we also consider the indices of the local maxima. Let the indices of selected sample points of the force curves $\mathbf{f}_k$ be stored in the training sets. And the local maxima indices are stored in the set B_k. Then for the k^{th} signal, the reduced data is given by the indices stored in the set $S_k = A_k \cup B_k$. We hypothesize that the adaptive sampling provides the informative features for modeling the deformable objects, which will be validated through the trained models.

In Fig. 5.2, we show the results of the approach on three typical force curves for the Weber fraction $\delta = 0.05$. The red circles in the figure are the selected sample points. As illustrated in the figures, the approach is can handle the complexities (i.e, non-linearity and rate-dependent behavior) of visco-elastic deformable objects. For example, when the signal is changing rapidly (slowly), the approach is bound to select more (fewer) samples. The Weber fraction δ determines the indices of the selected samples and hence, the reduced data size.

In our data, we select the $\delta = 0.05$ considering the threshold of the force perception of the finger. This threshold selects about 4% of the training data throughout the materials. This data selection method is proportion to the amplitude of the training signal, as shown in Table 5.1.

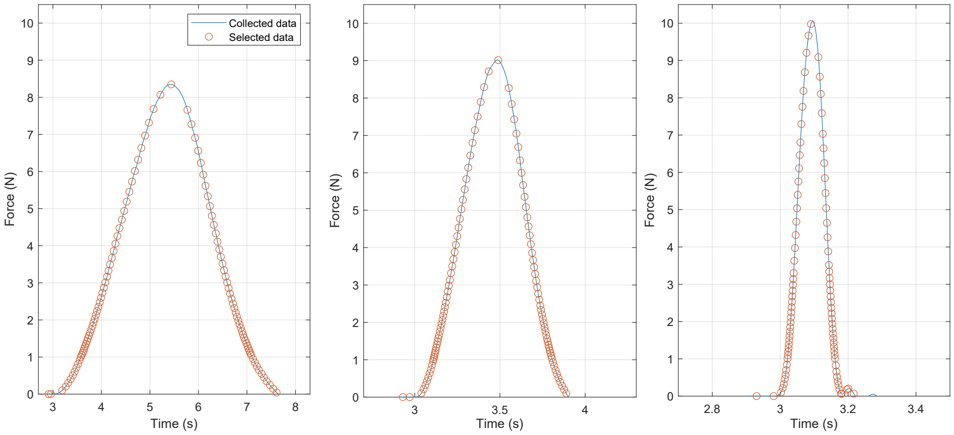


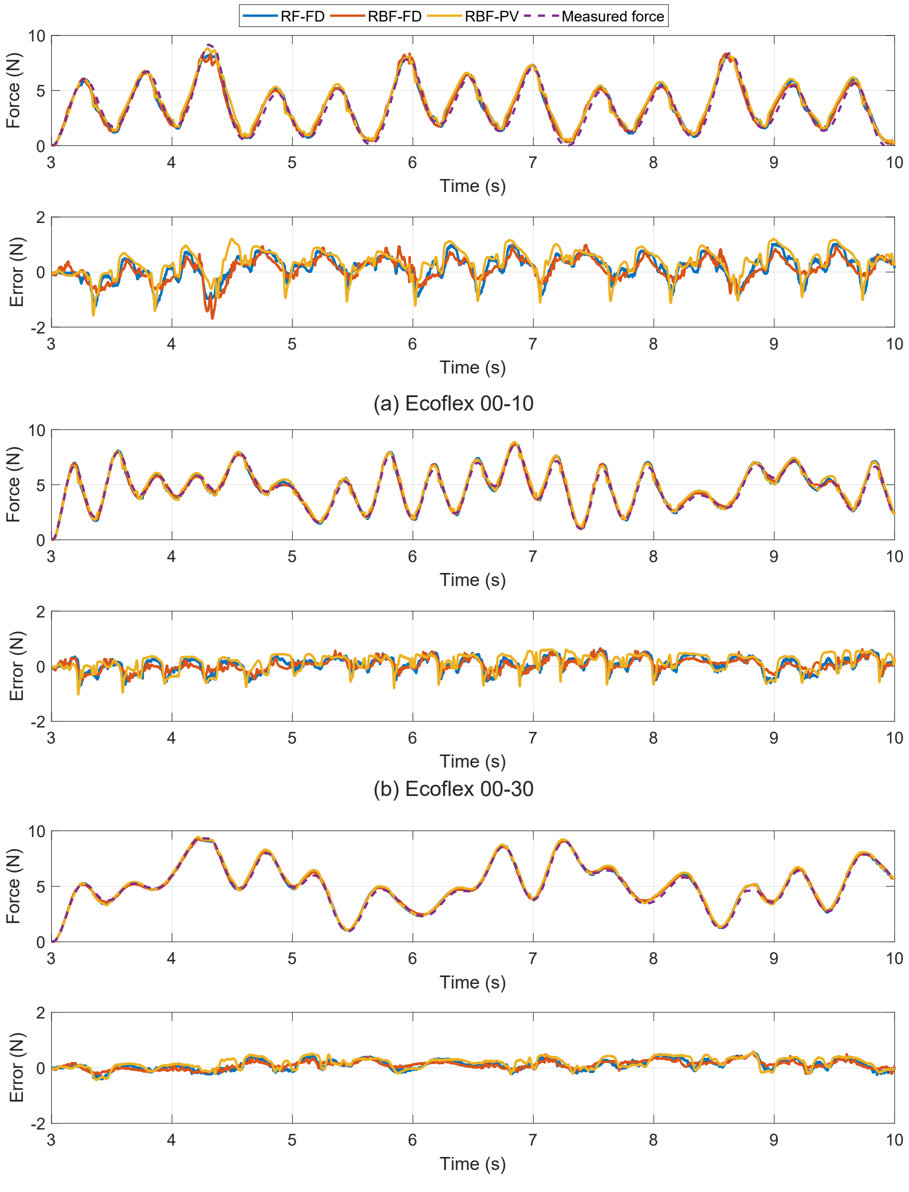
Figure 5.2: Examples of the collected data and the selected samples. The applied perceptual sampling points are shown in three different frequencies where the threshold δ is 0.05. The lowest frequency signal example, 0.2 Hz is on the leftmost figure, and the highest frequency signal example, 5 Hz is on the rightmost figure. The middle figure shows the example of 1 Hz frequency signal.

5.2 Modeling performance

We train the viscoelastic models to validate the effectiveness of the proposed perceptual sampling method. The modeling method is the same as in Section 4.3: RF-FD, RBF-FD, and RBF-PV. We add a suffix, PS, to the name of the trained model for clarity. We also train the models of random sampling using 4% of the data. These models are only used for fair comparison in modeling because of their instability in rendering in the preliminary test.

We group the two materials, Ecoflex 00-50 and Dragon Skin FX, as Ecoflex 00-30. It is because modeling results of Ecoflex 00-30, Ecoflex 00-50, and Dragon Skin FX, showed similar properties considering Table 4.2.

Figure 5.3 compares the modeling results on the test set in the exemplar force curves predicted by the perceptual sampling models. Compared to the random sampling results, the perceptual sampling models show lower error rates in the stiffer ma-

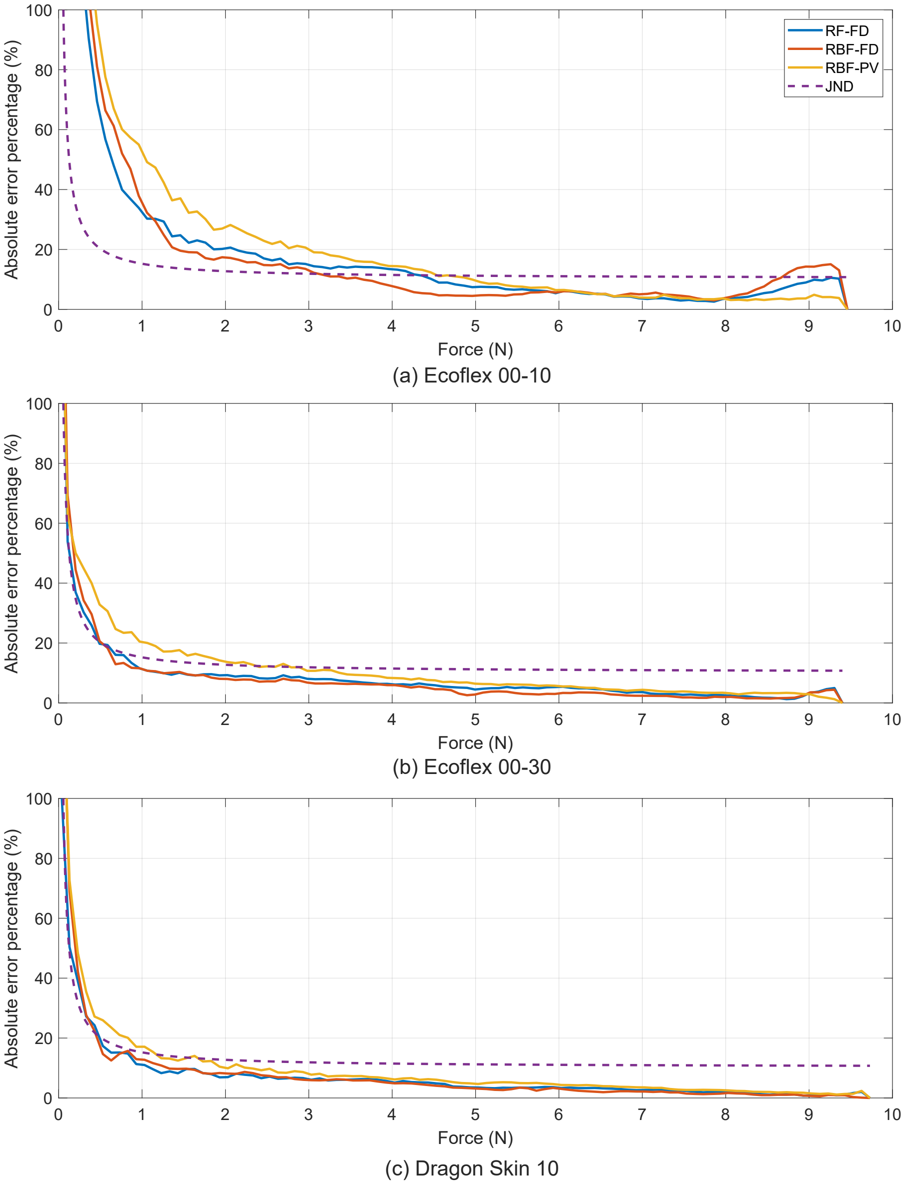


(a) Ecoflex 00-10

(b) Ecoflex 00-30

(c) Dragon Skin 10

Figure 5.3: Exemplar modeling results on test signals. We show the results from the same test set with Figure 4.6.



(a) Ecoflex 00-10

(b) Ecoflex 00-30

(c) Dragon Skin 10

Figure 5.4: Exemplar modeling results on test signals.

Table 5.1: Average number of the selected data point of each signal by maximum amplitude

Material	2 N	5 N	8 N
Ecoflex 00-10	64.44	104.78	124.89
Ecoflex 00-30	65.67	101.44	120
Dragon Skin 10	68.44	104.89	123

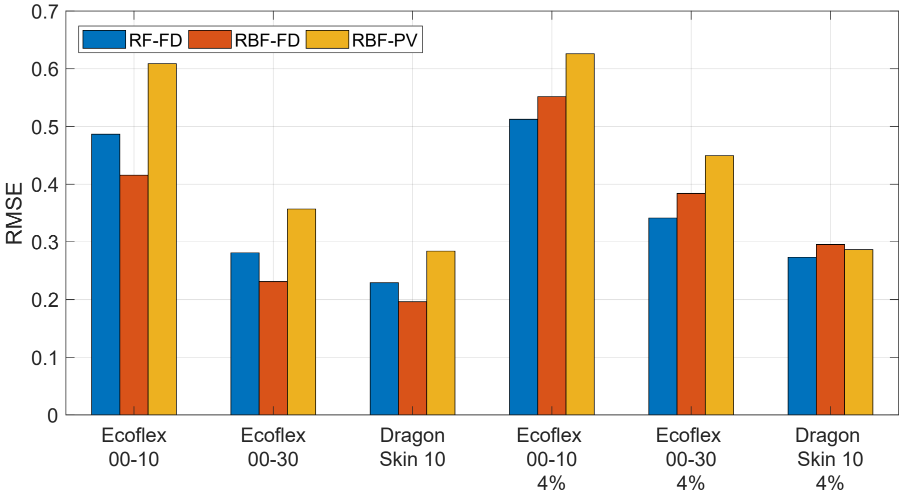


Figure 5.5: RMSEs of the three trained models with two sampling methods.

terials, Dragon Skin 10. The soft material, Ecoflex 00-10, shows very similar results as the previous modeling results. These results show that our perceptual sampling method is more effective on stiffer materials.

Figure 5.4 shows the absolute percentage error (APE) by the reference force. As described in Section 4.3.3, most models' APEs lie below the JND curves except for the two FD models on 9 N force. This point is also depicted in the modeling results, such as 4.3 seconds in Figure 5.3 (a). The training samples from the perceptual sampling are insufficient to train the fractional derivative models on the largest force point. This point can be improved, by using a smaller threshold to collect dense points near local maximum points or using a different training set rather than the sinusoidal signals.

Figure 5.5 shows the average RMSE results of the test set. Here, we also use the 4%, as mentioned earlier, random-sampled model to compare in error rate and modeling time. The RMSE results of all trained models show a big difference between the sampling method. The large differences are shown in the RBF-FD models. Regarding the modeling time in Figure 5.6, we can see the significant difference between the two sampling methods. The modeling times of perceptual sampling show almost consistent regardless of the models, while the RBF models take about 16 times longer. The well-organized samples increase the efficiency of modeling.

The modeling results of perceptual sampling models indicate that all models are improved in both RMSE and modeling time. Regarding the modeling time, the RBF models have a big advantage over the perceptual sampling method. The huge modeling time gaps in the sampling method come from the well-organized sampling because it helps to minimize the calculation to find the best 100 kernels. This feature equalizes the modeling time of the RBF models with the RF model. The RMSE values of the RBF-FD model between the two sampling methods are also noticeable. The RBF-FD models mark the lowest error rate among the trained models in all materials, which shows that the RBF-FD benefits from the sampling method.

5.3 Perceptual experiment of perceptual sampling

We designed the perceptual experiment to compare the two different sampling methods. Considering the first experiment's results, we focus on comparing the two sampling methods with the modeling methods. The overall process is similar to the first experiment in Section 4.5.

5.3.1 Improved haptic rendering

In this experiment, we adapt precise haptic rendering for the random forests using the first-order estimation of past positions. The previous haptic rendering method queue the device position to calculate the fractional derivatives. This method is accu-

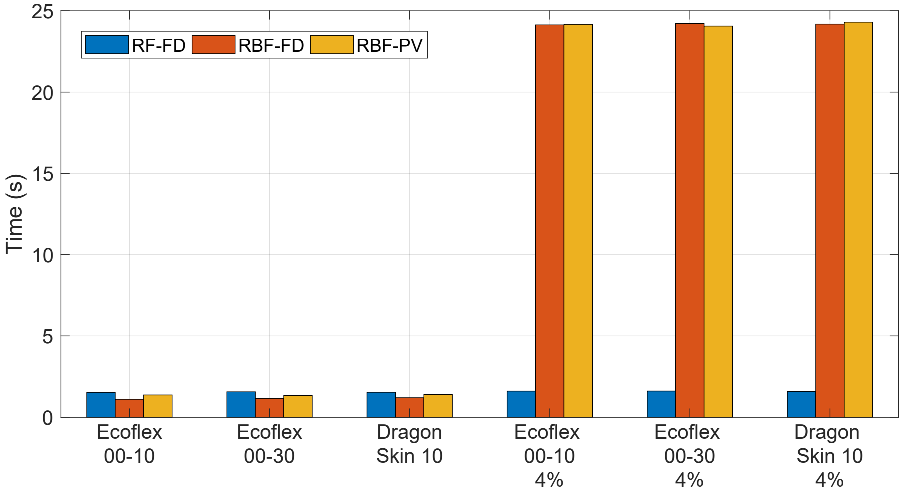


Figure 5.6: Modeling time of the three trained models with two sampling methods.

rate if the positions' intervals are equal to 1 ms. However, the interval's variety affects the fractional derivatives' estimation. It gets even worse when the update rate becomes lower.

Here, we use first-order position estimation of the recorded trajectories using the device position and the timestamp. We queue the position and the timestamp until the difference between the first and the last is n ms. Then, we estimate the positions of the past n seconds. The estimated positions are directly used as input to the fractional derivatives. This method is well-optimized in that it does not highly affect the update rate.

5.3.2 Methods

Participants

We recruited 20 adults (6 females and 14 males, average age of 23.0 years) for this experiment. None of them had a history of neurophysiology disorders. They were provided with written and verbal instructions about the experiment. They were paid KRW 25,000 ($\simeq$ USD 19) for compensation.

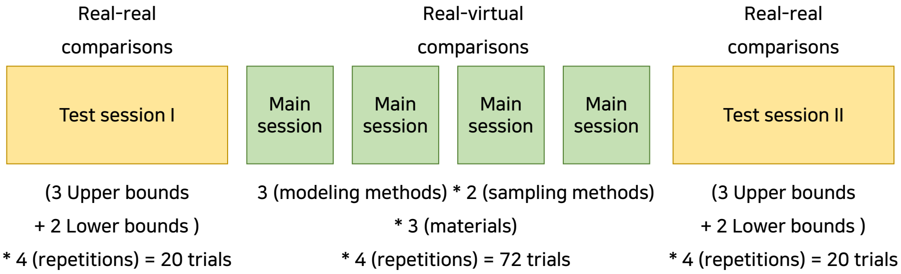


Figure 5.7: Experiment procedures of the perceptual sampling method.

Task and conditions

The overall task is similar to the previous experiment (see Section 4.5 for the figures). The comparison task is the same as before while the experiment conditions are changed.

We tested 18 experiment conditions: the three deformable objects (Ecoflex 00-10, Ecoflex 00-30, and Dragon Skin 10) × the three modeling methods (RF-FD, RBF-FD, and RBF-PV) × the two sampling methods (RS and PS). Then, we added the perceptual sampling conditions so that a participant evaluates six different trained models per material.

Then, we also prepared the upper-bound and the lower-bound conditions. The upper-bound conditions include real object pairs of three materials. The two most apparently different objects represented the lower-bound conditions. For the balancing, we repeated lower-bound conditions twice as a soft-hard condition and a hard-soft condition from left to right.

Procedures

The experiment consisted of three steps in sequence: Test session I, the Main session, and Test session II, as depicted in Figure 5.7. In test sessions I and II, we placed the comparisons between the real objects, while the main session only included the real-virtual comparison. In each comparison, participants compared two deformable objects, as explained in Section 4.5.

In a single repetition of the test sessions, the three upper-bound conditions and the two lower-bound conditions were given. The order of the five conditions was randomized and repeated four times. Thus, the total number of each test session was 20. Test session I worked as a guideline for similarity scores for participants. The test session II, which was given at the last of the experiment, was for verifying their similarity scores.

In the main session, all 18 experiment conditions were tested. These conditions were shuffled and repeated four times. The main session includes 72 trials. Each participant completed all four sessions in a within-subject design.

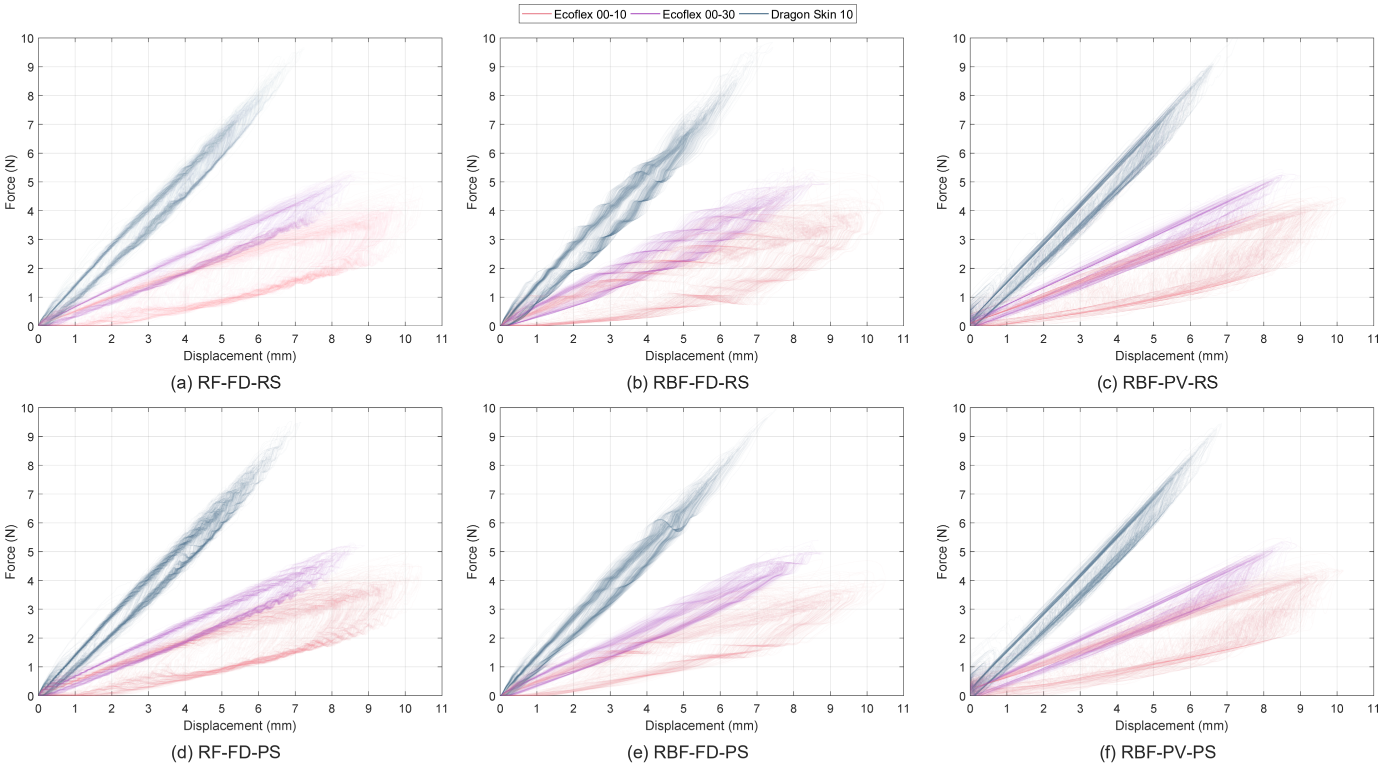
After the experiment, we also asked the same five questions as in the questionnaire in Section 4.5.

5.3.3 Results

HIP Trajectories

Figure 5.8 shows the recorded trajectories of HIP and the resultant force of the model during the experiment when the interaction happens. We removed about 20% percent of the data, which penetrates the visual guideline because it corrupts the shape of hysteresis curves. As shown in Figure 4.18, these hysteresis curves of the trained models represent the model behaviors.

For all models, the perceptual sampling shows the concrete rendering shapes than the random sampling. The rendering trajectories of PS are more likely to follow certain lines. Unlike the modeling RBF-PV method, the behaviors of two fractional derivative models, RF-FD and RBF-FD, are slightly changed following the sampling method. The RBF-FD-PS model shows a better shape of hysteresis curves than the RBF-FD-RS. However, the RF-FD-PS model shows small wave patterns over the curves.



(a) RF-FD-RS

(b) RBF-FD-RS

(c) RBF-PV-RS

(d) RF-FD-PS

(e) RBF-FD-PS

(f) RBF-PV-PS

Figure 5.8: Displacement-force hysteresis curves constructed from the HIP trajectories of all participants.

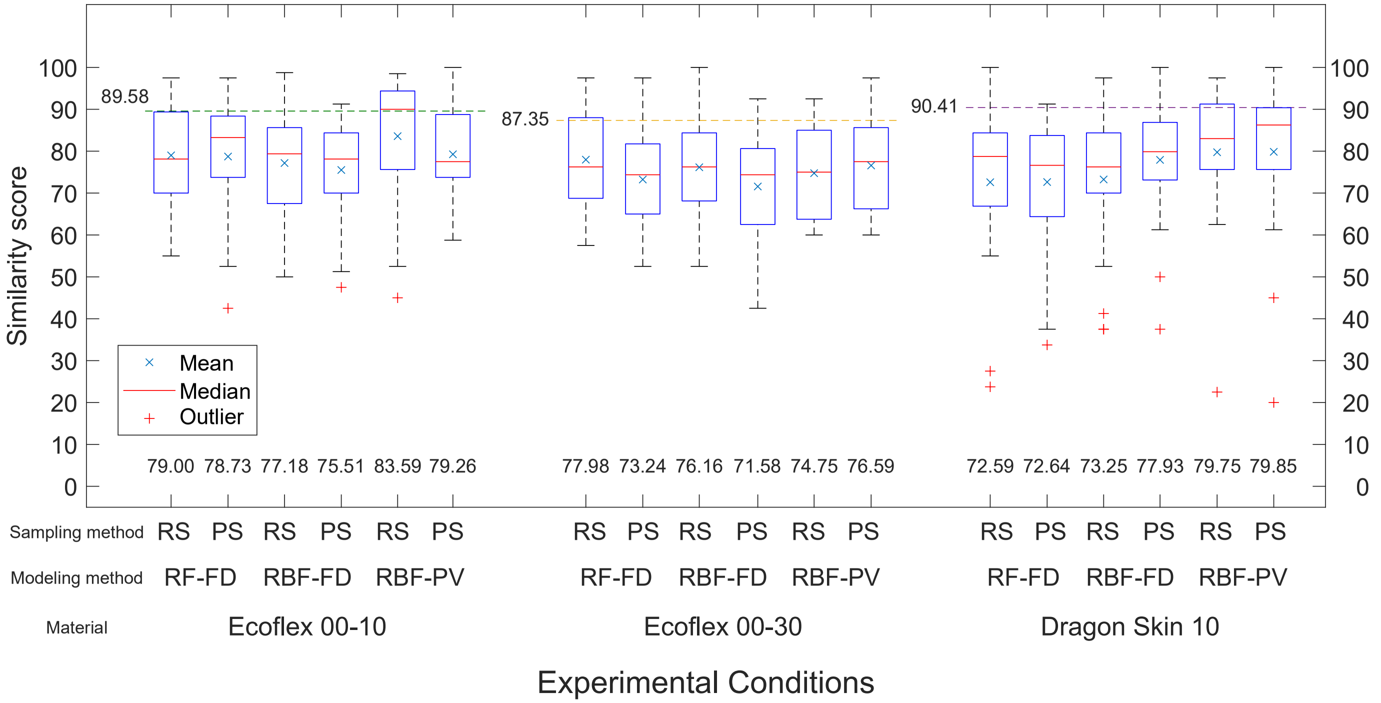


Figure 5.9: Similarity rating results of the perceptual experiment. The means of each condition are specified right above the horizontal axis. For each material, conditions without statistically significant differences are grouped using the same letters.

Similarity Scores

Figure 5.9 presents the collected similarity scores SS in box plots. For the analysis, we averaged the answered similarity scores for all 18 experimental conditions in the main session. We also averaged the upper-bound condition and the lower-bound conditions in the test session II. As we split the upper-bound condition from the main experiment, the average similarity score of each material is shown over the dotted line. The box plots are grouped in three by materials. Then, the modeling methods are placed one by one. Two sampling methods of one modeling method are placed side by side. The median value of each model is shown in a red horizontal line, and the mean value is represented as a blue mark. For the three materials, the RF-FD-PS models resulted in the mean SS values between 72.64 and 78.73, the RBF-FD-PS between 71.58 and 77.93, and the RBF-PV-PS 76.59 and 79.85.

For statistical analysis, we applied three-way repeated measures ANOVA with the independent variables of $Sampling\,method$ (RS, PS), $Modeling\,method$ (RF-FD, RBF-FD, and RBF-PV), and $Material$ (Ecoflex 00-10, Ecoflex 00-30, and Dragon Skin 10) to SS. The lower-bound and upper-bound conditions were not included. Two main conditions, $Modeling\,method$ ($F(2, 38) = 4.485, p = 0.018$) and $Sampling\,method$ ($F(1, 19) = 4.973, p = 0.038$) have a significant effect. But, their interaction ($F(2, 38) = 0.419, p = 0.661$) has not a significant effect. In other hands, $Materials$ ($F(2, 38) = 1.539, p = 0.228$) condition is not significant and its interactions with the two other conditions, the modeling method $F(2, 38) = 2.576, p = 0.089$ and the sampling method $F(2, 38) = 2.576, p = 0.089$, are also not significant. The three-way interaction of the main factors $F(4, 76) = 2.687, p = 0.037$ shows the significant effect.

5.3.4 Discussions

The random sampling method is statistically better in similarity than the perceptual sampling considering the ANOVA and the post-hoc analysis results. The rendering quality decreases due to using five times smaller training samples. To show more detailed results of this comparison, we suggest a small human experiment to gather a detailed description of each model.

The ANOVA results also show that the results mostly depend on the interaction of all three factors. Two effects, the modeling and the sampling methods, are statistically significant, but their interaction is only significant when used with the materials. This phenomenon means that all the proposed methods are strong in some specific situations. For example, the RBF-PV-RS model shows the best performance in Ecoflex 00-10, while the RF-FD-RS model is the best in Ecoflex 00-30.

There is some outlier, mostly in the Dragon Skin 10 material. Most participants also reported that they felt some vibrations in some conditions. We also noticed some irregular unintended vibration-like effects during the experiment for some specific participants. It mostly happens under the lower update rates. As depicted in Figure 5.8, the perceptual sampling method of the RF-FD model shows more vibration-like movement than the random sampling one. This vibration affects the stiffer object like Dragon Skin 10 more than the soft material like Ecoflex 00-10. This happens because the sampled points were similar in every signal. Sampling on more complex signals would help to train a better RF-FD model.

We can also design an experiment on how the percentages of the perceptual sampling affect the actual perceptual feeling. The relationship between the perceptual threshold and the actual feeling can guide the optimization methods.

Among the two models we proposed, RF-FD showed the best appearance in modeling due to the advantages in both error and modeling time. However, in the case of generating force directly to a person in rendering, tremors peculiar to the random forest model sometimes occur, or the time required to search the tree is too tight to be done within 1 ms, so the update rate sometimes suffers a loss, which causes similarity scores to be lowered. To sum up, our FD methods gain in the modeling, but the rendering results could be better due to the problems inherent in the model even though rendering shows acceptable quality overall.

Training the RBF-FD model in perceptual sampling is meaningful because it shows sufficient accuracy and resolution in rendering using only two fractional orders. The rendering sometimes looks better for the same object compared to RF.

VI. Conclusions

6.1 Conclusion

We propose modeling and rendering methods on viscoelastic materials using fractional derivatives as an input function for the first time. We use the methods of two non-linear mapping functions, the random forests and the radial basis functions. For the random forests, we use ten different orders of fractional derivatives and apply a perceptual sampling method. For the RBFs method, we select two fractional orders of 0.1 and 0.2 to train the model. We evaluate our models using objective measurement and subjective assessment with four different materials. We compare the measured data with our two fractional derivative models and one reference RBFs model from the previous studies. It shows that two fractional derivatives models outperform the reference model, which uses position and velocity as a feature vector. Most participants who compared the similarities between the real objects and the virtual models reported that it was indistinguishable.

We also improve the modeling methods using the perceptual sampling method. This perceptual sampling method is motivated by the well-known adaptive sampling. We train and evaluate the modeling methods with the same parameters as the random sampling method. The result shows that all models are improved in error rates. The two RBF methods have more benefits from the sampling method. The human experiment also shows that the perceptual sampling methods show similar results to the previous random sampling methods.

6.2 Future works

As we discussed in the modeling and rendering part, there is room to find better fractional orders for expressing each material, even though the random forest methods can find the best fit inside the model. For example, a narrower range of the fractional derivative order, such as $r = 0.01$ to 0.20 for the Dragon Skin materials, can represent a more detailed material property. This optimization is left for future work.

We also can optimize the RBF-FD model features. In this dissertation, we limited the number of features to two for a fair comparison with the RBF-PV model. This feature number can be expanded for precise modeling. As a guideline for future work, we tested the modeling and rendering of the RBF-FD method of ten fractional order feature vectors like the RF-FD model, which was unstable in rendering at some conditions.

We may apply this concept to other time-dependent components, such as acceleration, considering that we use a fractional derivative as an alternative to the velocity feature. Our first idea is driven by the fact that the number of the feature.

Our data-driven modeling method of fractional derivatives has been used on the viscoelastic behavior highly correlated to both position and velocity. Our research performs a primitive work that uses fractional derivative features in the limited environment of rendering rates of 1 kHz. Using better estimators with fractional derivatives or more exact fractional orders are open research questions. For example, an adaptive order selection is applied to model image-based texture samples in a recent study [19]. Detailed-ordered features like a fractional order of 0.01 or a better estimator can improve the non-linear estimator.

요 약 문

우리는 일상의 다양한 물체와 시각적인 인지 외에도 직접 손으로 닿는 등 상호작용을 하면서 살아간다. 이와 같이 가상의 공간에서도 실재감을 높이고 현실과 유사한 촉감을 제공하기 위해, 실체 촉감과 유사한 상호작용 모델을 만들고 렌더링하는 연구가 진행되었다. 그 중에서도 인간의 피부의 성질과도 밀접한 연관성이 있는 점탄성은 물질 자체의 복잡성 때문에 데이터 기반 모델링 방법을 사용했지만, 모델링에 많은 데이터를 써야하고 모델링 시간이 오래 걸리는 등 현실적인 장벽이 있었다.

본 논문에서는 이를 해결하기 위해 데이터 기반 점탄성 모델링에 분수계 미분을 도입하였고, 이에 대한 결과를 다룬다. 기존의 모델에서는 점탄성의 물리적 의미를 고려하여 탄성의 위치, 점성의 속도, 그리고 속도의 저주파 통과 필터 값을 사용하여 모델링을 진행하였다. 우리는 여기서 위치를 1차 미분한 값인 속도와 0차 미분한 위치 사이의 값을 분수계 미분 방법을 통해 더욱 세분화하여 모델에 활용하였다. 여기서 사용한 분수계 미분 방법은 그룬월드-레트니코브 분수계 미분을 촉각 모델링 및 렌더링에 최적화한 것으로, 디지털 신호로 전환할 때 양자화되는 시간과 컴퓨터 공학에서의 계산적 한계를 고려하여 단순화하였다. 여기에 본 연구에서 설계한 점탄성 모델을 고려하여 필요한 수치를 미리 선정한 후 필터의 형태로 계산하도록 하여 모델링 및 렌더링에 활용하였다.

우리는 이렇게 만들어둔 분수계 미분 값을 토대로 점탄성 상호작용 중 균일한 물질에 대한 1차원의 힘에 대해서 랜덤 포레스트와 방사 기저 함수 방법을 통해

모델링 및 렌더링을 먼저 진행하였다. 랜덤 포레스트 회귀 분석에 10개의 분수계 미분값을 사용하여 점탄성 모델을 생성하였다. 또한, 방사 기저 함수 보간 방법에는 랜덤 포레스트 회귀 분석에 사용한 미분 값 중 가장 유효한 2개의 분수계 미분값을 각각 물체마다 적용하여 점탄성 모델링을 수행하였다. 모델링 결과로는 실제 측정한 결과 값과 모델들의 예측 값 사이의 에러가 인간이 인지가능한 최소식별차 이내로 나왔다. 이후 렌더링 결과를 확인하기 위해 동일한 조건 하에서 실제 물체 및 가상의 물체와 상호작용을 진행하고 인지적 유사도를 측정하였다. 분수계 미분을 사용한 모델링 방법이 기존의 방법에 비해 실제 물체에 더 유사하면서, 실제 물체와 구분이 없을 정도의 안정적인 모델 구현을 보여주었다.

이후 본 연구에서는 이 1차원의 힘에 대한 모델링 방법을 더욱 최적화하기 위해서 샘플링 방법을 랜덤한 데이터대신 촉감의 인지적 한계점을 바탕으로 추출한 데이터를 사용하여 모델링 및 렌더링을 수행하였다. 새로운 샘플링 방법은 기존 모델에서 중복으로 활용되는 부분이 줄어들면서 훨씬 더 적은 시간과 낮은 수준의 에러 비율을 확보하는 성과를 거두었다. 다만 이를 사용자 실험을 통해 평가하였을 때는 사람들은 인지적으로 랜덤 데이터 추출과 이번에 제안한 인지 기반 모델링 방법을 구분할 수 있었다. 이는 주로 일부 모델에서의 불안정한 렌더링을 보였고, 렌더링 결과상 랜덤 포레스트에 최적화되지 않은 점을 보였기 때문으로 보인다. 추후에 학습시에 더 복잡한 신호를 사용하거나 안정적인 렌더링을 제공할 경우 인지적으로 차이가 없을 정도로 유사한 모델을 제공할 수 있을 것으로 예상한다.

References

[1] Susan J Lederman and Klatzky Roberta L. Hand movement: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368, 1987.

[2] Amit Bhardwaj, Hojun Cha, and Seungmoon Choi. Data-driven haptic modeling of normal interactions on viscoelastic deformable objects using a random forest. *IEEE Robotics and Automation Letters*, 4(2):1379–1386, 2019.

[3] Yuan cheng Fung. *Biomechanics: Mechanical properties of living tissues*. Springer Science & Business Media, 2013.

[4] Andrew H. C. Gosline and Vincent Hayward. Dual-channel haptic synthesis of viscoelastic tissue propertiesusing programmable eddy current brakes. *International Journal of Robotics Research*, 28:1387–1399, 2009. ISSN 0278-3649. doi: 10.1177/0278364909105274.

[5] Mert Sedef, Evren Samur, and Cagatay Basdogan. Real-time finite-element simulation of linear viscoelastic tissue behavior based on experimental data. *Virtual and Augmented Reality Supported Simulator*, 26:58–68, 2006. ISSN 0272-1716. doi: 10.1109/mcg.2006.135.

[6] G. W. S. Blair and F. M. V. Coppen. The subjective judgement of the elastic and plastic properties of soft bodies; the ”differential thresholds” for viscosities and compression moduli. *Proceedings of the Royal Society of London*, 128(850): 109–125, 1939.

[7] G. W. S. Blair and F. M. V. Coppen. The subjective conception of the firmness of soft materials. *American Journal of Psychology*, 55(2):215–229, 1942. ISSN 0002-9556. doi: 10.2307/1417080.

[8] G. W. S. Blair and F. M. V. Coppen. The estimation of firmness in soft materials. *The American Journal of Psychology*, pages 234–246, 1943.

[9] G. W. Scott Blair. Analytical and integrative aspects of the stress-strain-time problem. *Journal of Scientific Instrument*, 21:80–84, 1944. ISSN 0950-7671. doi: 10.1088/0950-7671/21/5/302.

[10] G.W.S. Blair, B. C. Veinoglou, and J. E. Caffyn. Limitations of the newtonian time scale in relation to non-equilibrium rheological states and a theory of quasi-properties. *Proceedings of the Royal Society of London Series A. Mathematical and Physical Sciences*, 189(1016):69–87, 1947. ISSN 0080-4630. doi: 10.1098/rspa.1947.0029.

[11] R. L. Bagley and P. J. Torvik. A theoretical basis for the application of fractional calculus to viscoelasticity. *Journal of Rheology*, 27:201–210, 1983. ISSN 0148-6055. doi: 10.1122/1.549724.

[12] Ronald L. Bagley and Peter J. Torvik. Fractional calculus-a different approach to the analysis of viscoelastically damped structures. *AIAA Journal*, 21(5):741–748, May 1983. ISSN 0001-1452. doi: 10.2514/3.8142.

[13] Richard L. Magin. Fractional calculus models of complex dynamics in biological tissues. *Computers and Mathmatics with Applications*, 59:1586–1593, March 2010. ISSN 0898-1221. doi: 10.1016/j.camwa.2009.08.039.

[14] A. Bonfanti, J. L. Kaplan, G. Charras, and A. Kabla. Fractional viscoelastic models for power-law materials. *Soft Matter*, 16:6002–6020, 2020. ISSN 1744-683X. doi: 10.1039/d0sm00354a.

[15] M. Sasso, G. Palmieri, and D. Amodio. Application of fractional derivative models in linear viscoelastic problems. *Mechanics of Time-Dependent Materials*, 15: 367–387, 2011. ISSN 1385-2000. doi: 10.1007/s11043-011-9153-x.

[16] F.C. Meral, T.J. Royston, and R. Magin. Fractional calculus in viscoelasticity: An experimental study. *Communications in Nonlinear Science and Numerical Simulation*, 15(4):939–945, May 2010. doi: 10.1016/j.cnsns.2009.05.004.

[17] Lloyd B. Eldred, William P. Baker, and Anthony N. Palazotto. Kelvin-voigt vs fractional derivative model as constitutive relations for viscoelastic materials. *AIAA Journal*, 33(3):547–550, March 1995. doi: 10.2514/3.12471.

[18] Ozan Caldiran, Hong Z. Tan, and Cagatay Basdogan. Visuo-haptic discrimination of viscoelastic materials. *IEEE Transactions on Haptics*, 12(4):438–450, 2019. ISSN 1939-1412. doi: 10.1109/toh.2019.2924212.

[19] Huiran Hu and Aiguo Song. Haptic texture rendering of 2d image based on adaptive fractional differential method. *Applied Sciences*, 12(23):12346, dec 2022. doi: 10.3390/app122312346.

[20] Francisco J. Montáns, Francisco Chinesta, Rafael Gómez-Bombarelli, and J. Nathan Kutz. Data-driven modeling and learning in science and engineering. *Comptes Rendus Mécanique*, 347(11):845–855, nov 2019. doi: 10.1016/j.crme.2019.11.009.

[21] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.

[22] Peter W. Hatfield, Jim A. Gaffney, Gemma J. Anderson, Suzanne Ali, Luca Antonelli, Suzan Başeğmez du Pree, Jonathan Citrin, Marta Fajardo, Patrick Knapp, Brendan Kettle, Bogdan Kustowski, Michael J. MacDonald, Derek Mariscal, Madison E. Martin, Taisuke Nagayama, Charlotte A. J. Palmer, J. Luc Peterson, Steven Rose, J. J. Ruby, Carl Shneider, Matt J. V. Streeter, Will Trickey, and Ben Williams. The data-driven future of high-energy-density physics. *Nature*, 593:351–361, May 2021. ISSN 0028-0836. doi: 10.1038/s41586-021-03382-w.

[23] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, feb 2019. doi: 10.1016/j.jcp.2018.10.045.

[24] Mattia Cenedese, Joar Axås, Bastian Bäuerlein, Kerstin Avila, and George Haller. Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds. *Nature Communications*, 13(1), feb 2022. doi: 10.1038/s41467-022-28518-y.

[25] Robert Pollice, Gabriel dos Passos Gomes, Matteo Aldeghi, Riley J. Hickman, Mario Krenn, Cyrille Lavigne, Michael Lindner-D'Addario, AkshatKumar Nigam, Cher Tian Ser, Zhenpeng Yao, and Alán Aspuru-Guzik. Data-driven strategies for accelerated materials design. *Accounts of Chemical Research*, 54 (4):849–860, feb 2021. doi: 10.1021/acs.accounts.0c00785.

[26] Azfar Inteha, Nahid-Al-Masood, Farhan Hussain, and Ibrahim Ahmed Khan. A data driven approach for day ahead short term load forecasting. *IEEE Access*, 10: 84227–84243, 2022. doi: 10.1109/ACCESS.2022.3197609.

[27] FX Nugroho Soelami, Putu Handre Kertha Utama, Irsyad Nashirul Haq, Justin Pradipta, Edi Leksono, and Meditya Wasesa. Data driven building electricity consumption model using support vector regression. *Journal of Engineering and Technological Sciences*, 53(3):210313, jul 2021. doi: 10.5614/j.eng.technol.sci. 2021.53.3.13.

[28] Sibo Cheng, I. Colin Prentice, Yuhan Huang, Yufang Jin, Yi-Ke Guo, and Rossella Arcucci. Data-driven surrogate model with latent data assimilation: Application to wildfire forecasting. *Journal of Computational Physics*, 464:111302, sep 2022. doi: 10.1016/j.jcp.2022.111302.

[29] Karon E MacLean. The 'haptic camera': A technique for characterizing and

playing back haptic properties of real environments. *Proc. of Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAPTICS)*, pages 459–467, 1996.

[30] Stephanie Greenish, Vincent Hayward, Vanessa Chial, Allison Okamura, and Thomas Steffen. Measurement, analysis, and display of haptic signals during surgical cutting. *Presence: Teleoperators & Virtual Environments*, 11(6):626–651, 2002.

[31] Allison M Okamura, Robert J Webster, Jason T Nolin, KW Johnson, and H Jafry. The haptic scissors: Cutting in virtual environments. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 828–833, 2003.

[32] Timothy Edmunds and Dinesh K Pai. Perceptual rendering for learning haptic skills. In *Proc. of Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS)*, pages 225–230, 2008.

[33] Mark B Colton and John M Hollerbach. Reality-based haptic force models of buttons and switches. In *IEEE International Conference on Robotics and Automation*, pages 497–502, 2007.

[34] Sheldon Andrews and Jochen Lang. Interactive scanning of haptic textures and surface compliance. In *Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 99–106, 2007.

[35] Christopher Richard, Mark R Cutkosky, and Karon MacLean. Friction identification for haptic display. *Proc. of Haptic Interfaces for Virtual Environments and Teleoperator Systems (HAPTICS)*, pages 14–19, 1999.

[36] Jochen Lang and Sheldon Andrews. Measurement-based modeling of contact forces and textures for haptic rendering. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):380–391, 2011.

[37] Joseph M. Romano and Katherine J. Kuchenbecker. Creating realistic virtual textures from contact acceleration data. *IEEE Transactions on Haptics*, 5(2): 109–119, 2012.

[38] S. Shin and S. Choi. Geometry-based haptic texture modeling and rendering using photometric stereo. In *IEEE Haptics Symposium (HAPTICS)*, pages 262–269, 2018.

[39] Reza Haghighi Osgouei, Sunghwan Shin, Jin Ryong Kim, and Seungmoon Choi. An inverse neural network model for data-driven texture rendering on electrovibration display. In *IEEE Haptics Symposium (HAPTICS)*, pages 270–277, 2018.

[40] Hyejin Choi, Seongwon Cho, Sunghwan Shin, Hojin Lee, and Seungmoon Choi. Data-driven thermal rendering: An initial study. In *IEEE Haptics Symposium (HAPTICS)*, pages 344–350, 2018.

[41] Dinesh K. Pai, Jochen Lang, John Lloyd, and Robert J. Woodham. Acme, a telerobotic active measurement facility. In *In Proceedings of the Sixth International Symposium on Experimental Robotics VI*, 1999.

[42] Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. Scanning physical interaction behavior of 3d objects. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 87–96, August 2001. doi: 10.1145/383259.383268.

[43] Raphael Hover, Gábor Kósa, Gabor Szekly, and Matthias Harders. Data-driven haptic rendering—from viscous fluids to visco-elastic solids. *IEEE Transactions on Haptics*, 2(1):15–27, 2009.

[44] Raphael Höver and Matthias Harders. Measuring and incorporating slip in data-driven haptic rendering. In *IEEE Haptics Symposium (HAPTICS)*, pages 175–182, 2010.

[45] Anatolii Sianov and Matthias Harders. Data-driven haptics: Addressing inhomogeneities and computational formulation. In *IEEE World Haptics Conference (WHC)*, pages 301–306, 2013.

[46] Sunghoon Yim, Seokhee Jeon, and Seungmoon Choi. Data-driven haptic modeling and rendering of viscoelastic and frictional responses of deformable objects. *IEEE Transactions on Haptics*, 9(4):548–559, 2016.

[47] Anatolii Sianov and Matthias Harders. Exploring feature-based learning for data-driven haptic rendering. *IEEE Transactions on Haptics*, 11(3):388–399, 2018. ISSN 1939-1412. doi: 10.1109/toh.2018.2817483.

[48] Arsen Abdulali, Ibragim R. Atadjanov, Seungkyu Lee, and Seokhee Jeon. Realistic haptic rendering of hyper-elastic material via measurement-based FEM model identification and real-time simulation. *Computers & Graphics*, 89:38–49, June 2020. ISSN 0097-8493. doi: 10.1016/j.cag.2020.04.004.

[49] Arsen Abdulali and Seokhee Jeon. Data-driven haptic modeling of plastic flow via inverse reinforcement learning. In *2021 IEEE World Haptics Conference (WHC)*, pages 115–120. IEEE, July 2021. doi: 10.1109/whc49131.2021.9517181.

[50] Abhishek Jaiantilal. randomforest-matlab, 2010. URL `https://code.google.com/archive/p/randomforest-matlab/`. Accessed: 2018-12-04.

[51] Peter Hinterseer, Sandra Hirche, Subhasis Chaudhuri, Eckehard Steinbach, and Martin Buss. Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Transactions on Signal Processing*, 56(2):588–597, 2008.

[52] Amit Bhardwaj, Subhasis Chaudhuri, and Onkar Dabeer. Design and analysis of predictive sampling of haptic signals. *ACM Transactions on Applied Perception (TAP)*, 11(4):1–20, 2014.

[53] Eckehard Steinbach, Matti Strese, Mohamad Eid, Xun Liu, Amit Bhardwaj, Qian Liu, Mohammad Al-Ja'afreh, Toktam Mahmoodi, Rania Hassen, Abdulmotaleb El Saddik, et al. Haptic codecs for the tactile internet. *Proceedings of the IEEE*, 107(2):447–470, 2018.

[54] Subhasis Chaudhuri and Amit Bhardwaj. Kinesthetic perception. *A Machine Learning Approach*, 2018.

[55] Tomáš Kisela. Fractional differential equationsand their applications. Master's thesis, BRNO UNIVERSITY OF TECHNOLOGY, 2008.

[56] Leo Breiman. *Machine Learning*, chapter 1, pages 5–32. Springer, 2001.

[57] Farrokh Janabi-Sharifi, Vincent Hayward, and Chung-Shin J. Chen. Discrete-time adaptive windowing for velocity estimation. *IEEE Transactions on Control Systems Technology*, 8(6):1003–1009, 2000. ISSN 1063-6536. doi: 10.1109/87.880606.

[58] Raphael Höver, Massimiliano Di Luca, and Matthias Harders. User-based evaluation of data-driven haptic rendering. *ACM Transactions on Applied Perception*, 8(1):1–23, 2010. ISSN 1544-3558. doi: 10.1145/1857893.1857900.

[59] Seokhee Jeon and Seungmoon Choi. Real stiffness augmentation for haptic augmented reality. *Presence: Teleoperators and Virtual Environments*, 20(4):337–370, 2011.

[60] Heather Culbertson, Juliette Unwin, and Katherine J. Kuchenbecker. Modeling and rendering realistic textures from unconstrained tool-surface interactions. *IEEE Transactions on Haptics*, 7(3):381–393, 2014. doi: 10.1109/toh.2014.2316797.

[61] Sunghwan Shin and Seungmoon Choi. Hybrid framework for haptic texture modeling and rendering. *IEEE Access*, 8:149825–149840, 2020. doi: 10.1109/ACCESS.2020.3015861.

[62] Seokhee Jeon and Seungmoon Choi. Haptic augmented reality: Taxonomy and an example of stiffness modulation. *Presence: Teleoperators and Virtual Environments*, 18(5):387–408, 2009.

[63] Arsen Abdulali, Ibragim Atadjanov, Seungkyu Lee, and Seokhee Jeon. Measurement-based hyper-elastic material identification and real-time fem simulation for haptic rendering. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, number 10, pages 1–10, 2019.

Acknowledgements

졸업하고 나간 여러 선배들과 먼저 곁을 떠나가는 동기와 후배들을 보면서 저에게도 이런 순간이 올 수 있을까 싶었습니다. 오랜 시간 동안 포항이라는 곳에 있으면서 언젠가는 배움의 과정을 마치고 더 멀리 나아가야지 하는 마음을 굳게 먹었지만, 시간만 빠르게 흘러가고 생각한대로 마음먹은대로 풀리지 않았던 기간이 많았고 또 길었습니다. 배움에는 끝이 없고 삶은 배움의 연속이라고 하더라도, 마침내 하나의 과정을 끝마치고 앞으로 나아갈 수 있게 되었습니다. 저의 길었던 배움의 과정 동안 늘 기다려주시면서 모범을 보이며 많은 것을 가르쳐주셨던 최승문 교수님께 먼저 한없는 감사의 말씀을 올립니다.

연구실에 처음 HCI를 하겠다고 당차게 들어온 후에 예상치 못하게 점탄성이라는 것에 눈독을 들이게 되었습니다. 가상의 물체를 만들어서 그 힘을 유사하게 표현해내는 것도 어쩌면 HCI의 일종이 아닐까 라고 스스로 되뇌이면서 고민하던 때에, 저에게 큰 힘이 되어주고 좋은 방향으로 이끌어준 Amit에게 깊은 감사를 드립니다. 특히, 무심코 던지고 지나칠뻔 했던 아이디어 덕분에 지금의 이 순간까지 이를 수 있게 되었습니다. 포항 땅에서 함께 연구하는 동안 여러 모델을 만들면서 많은 토론을 했고 지금은 인도에서 관련 연구를 계속 이어나가는 그에게서 정말 많은 도움을 받았고, 그렇기 때문에 그동안의 노력이 그에게도 많은 도움이 되었기를 바랍니다.

연구실에서 9년 가까운 시간을 보내면서 많은 대화를 나누었던 연구실 선후배들에게도 감사의 인사와 안부의 인사를 전하고 싶습니다. 저의 첫 연구실 사수였고 이게 멋있는 삶이구나 하는 것을 알려주셨던 재봉이형, 밤만 되면 많은 이야기를 나누었던 인이형, 같이 연구 아이디어를 짜내면서 장난도 많이 치기도 했던 건혁이형, 같은 기숙사 방에서 살며 재밌는 시간을 보냈던 종만이형, 문화가 달라도 많은 대화를 하며 많은 걸 배울 수 있었던 Reza, 서로 의사소통은 힘들었어도 잘 챙겨줬던 Phuong, 파이오니어 과제부터 마지막까지 정말 많은 것을 서슴없이 이야기할 수 있었던 호진이형, 처음엔 무서운줄 알았지만 따뜻하고 친근했던 성환이형, 오랜시간 알고 지내며 미운정고운정 다 쌓았던 용재형, 어려운 고비고비마다 항상 힘이

되어주셨던 승재형까지 모든 선배께서 보여주셨던 행동과 한 마디 한 마디 덕분에 졸업의 문턱을 넘어설 수 있게 되었습니다. 먼저 연구실을 떠나간 한슬이, 인석이, 종호형, 선웅이, 혜진이, 효승이, 범수, 민재, 진수 등 연구실 전 동료들에게도 같이 지내던 의미있는 시간들 덕분에 제가 여기까지의 여정을 이룰 수 있었다고 전하고 싶습니다. 현재 연구실을 이어가고 있는 상윤이, 겨레, 채용이, 지완이 등 함께 오랜 시간 보냈던 후배들과, 정은이, 다진이, 동근이, 정우, 재준이, 준우, 호석이, 그리고 들어온지 얼마 안 된 재혁이, 희연이까지 짧은 시간이라도 함께 할 수 있어서 든든했고 앞으로도 연구실에서 좋은 성과와 함께 많이 배워나갈 수 있기를 기원합니다.

마지막으로 본가에서 멀리 떨어져 나와서 낯선 땅에 자리를 잡고 공부를 시작할 때부터 돌아가는 순간까지도 항상 뒤에서 저를 위해 물심양면 지원해주시고 기도해주시고 응원해주신 부모님께 정말 감사드립니다. 어린 시절부터 많은 성공과 큰 성취만 겪어오다가 자연스럽게 겪게 되는 실패와 좌절 가운데서도 항상 믿어주시고 의지할 수 있어서 여기까지 올 수 있게 되었습니다. 또한, 말상대도 되어주고 같이 게임도 하면서 스트레스를 풀기도 했던 동생 혜준이에게도 감사하다는 인사를 전하고 싶습니다. 포항 땅에서 자주 보진 않아도 같은 땅덩어리 안에 있다는 것만으로도 편안함을 느낄 수 있었던 첫째, 셋째 큰아버지와 우리 사촌들에게도 항상 고맙다고 전하고 싶습니다.

Curriculum Vitae

Name : Hojun Cha

Education

2014. 08. – 2023. 02. Department of Computer Science and Engineering, Pohang University of Science and Technology (Ph.D.)

2009. 03. – 2014. 08. Department of Computer Science and Engineering, Pohang University of Science and Technology (B.S.)

Experience

2014. 08. – 2023. 02. Research Assistant, Interaction Laboratory, Department of Computer Science and Engineering, POSTECH, South Korea

2015. 09. – 2015. 12. Teaching Assistant, “Programming and Problem Solving”, Department of Computer Science and Engineering, POSTECH, South Korea

2018. 03. – 2018. 06. Teaching Assistant, “Objective-Oriented Programming”, Department of Computer Science and Engineering, POSTECH, South Korea

2017. 08. – 2020. 08. Technical Research Personnel, Alternative military service