2017 IEEE World Haptics Conference (WHC)
Fürstenfeldbruck (Munich), Germany
6–9 June 2017

Sample Selection of Multi-Trial Data for Data-Driven Haptic Texture Modeling

Arsen Abdulali, Waseem Hassan and Seokhee Jeon

Abstract— **In data-driven haptic texture rendering, the rendering quality is highly dependent on the quality of the input-output model training. The data in input model should be sufficient both in terms of quantity and coverage of the input space. Furthermore, the ever increasing input dimensions, to attain more realistic rendering makes the task of model building even more difficult. In order to address these problems, this paper proposes a novel sample selection algorithm. Our algorithm provides an efficient method of combining modeling data across multiple independent trials, whereby the significant model points are selected from each independent trial while the outliers are being eliminated. This study also provides a generic haptic model which equips other haptic modeling algorithms to benefit from the sample selection algorithm. The algorithm was evaluated using two isotropic and two non isotropic haptic texture datasets. The results showed that the algorithm provides upward of a two fold compression rate for model points, while at the same time the rendering quality remains unaffected.**

I. INTRODUCTION

Haptic texture rendering is an interactive process, where the tactile response is computed with respect to the user input, i.e., tactile interaction. This process is goverened by a predefined haptic texture model. The haptic texture model simulates the underling physics of contact [1] or blindly relates the input to the response based on observations from real interactions [2]. The latter approach for haptic texture modeling is referred to as the data-driven approach. This approach forgoes any knowledge of underlying physics or parameter tuning which is required in the former. Hence, the data-driven approach is considered as more applicable from an end user perspective.

The main aspects of a data-driven haptic texture model are the inputs and outputs of the system. The output can attain the form of modulated vibrations [2], [3] or friction [4], [5] based on the capabilities of the sensing organs,i.e. Meissner's and Pacinian corpuscles.

On the other hand, the input of the system can be very complex. Research on the input of data-driven model design started as a two-dimensional space, i.e., the velocity magnitude and normal force [3]. Such an input space was used to model isotropic haptic textures. Recently, anisotropic haptic textures were modeled by applying three-dimensional input [2], where the velocity magnitude was replaced by a two-dimensional velocity vector. Furthermore, this model has a flexible design that allows an increase in the input dimension. However, further increases of the input dimensions might make the data-driven approach impractical.

One of the weak aspects of any data-driven modeling technique is that the input space must be covered evenly by model points, whereas each model point must have an accurate feedback pattern, e.g., vibration signal. Sparse regions inside the input space can lead to significant reductions in the rendering quality, or even instability of the system. Hence, it can prove to be a challenging task to collect enough data to cover even a two-dimensional input space. Furthermore, if we increase the input space to three dimensions, covering the whole space can be a tedious task and can easily lead to an uneven covering of the space. One solution to this problem can be to gather data across multiple trials and concatenate it into a single model. In this case, every additional recording trial will contribute towards the model space coverage. However, a possible drawback can be that only a few points from each subsequent recording trial contribute to the model space, whereas the rest prove to be redundant. Additionally, another problem can be the number of outliers which might also increase with every additional recording trial.

This paper aims to drive the data-driven haptic texture modeling approach in a new direction by applying a novel concept of multi-trial data collection. The current study proposes an algorithm, which selects the representative set of model points from every trial and tries to remove the outliers. The resultant set of model points achieves the same level of rendering quality while providing the flexibility of increasing the input dimensions. Furthermore, a generic haptic texture model is also provided which ensures that the current algorithm can readily be used with any haptic texture modeling algorithm.

The rest of the paper is structured as follows. The background knowledge about haptic texture modeling and the sample selection in the field of haptics is provided in Sec. II. The proposed algorithm is described in Sec. III. In Sec. IV, the experimental setup and procedure is explained. The experimental results are reported in Sec. V, followed by conclusion and and future work in Sec. VI.

II. BACKGROUND

Initial efforts on haptic texture modeling originated from the texturing techniques in computer graphics, where the main goal was to create a spatial representation of the micro geometry of the surface. In [6], it was assumed that the micro geometry of the texture can be described by the change of the pixel magnitude of the gray scaled image of the surface. Thus, the haptic texture was reconstructed

The authors are with the Department of Computer Science and Engineering, Kyung Hee University, Yongin, Republic of Korea {abdulali, waseem.h, jeon}@khu.ac.kr

978-1-5090-1425-5/17/$31.00 ©2017 IEEE

by using this shading information in the form of a bump-map. Similarly, Minsky et al. proposed an approach, where spatially periodic patterns of the surface texture were used for a virtual spring perturbation [1]. This method was improved in [7] by applying a stochastic generator of force fields. These approaches were able to model perceptually describable surfaces, meaning that it was possible for user to point out at the surface which had been felt more or less rough. However, due to the lack of the real contact information in modeling, the feedback from these methods was unrealistic.

The contact information has significantly improved the rendering quality by applying data-driven modeling techniques, where the resultant model is trained based on the data that are collected during physical interaction with a real surface. Regardless of object properties and micro geometry of the surface, the data-driven model treats the user contact as a multidimensional input and maps it with the corresponding output, i.e., vibration or friction modulations.

In general, the data-driven modeling of haptic textures consists of three steps: data collection, data preprocessing and segmentation, and model building [2]. The data are usually collected during free exploration of the object surface or using a special recording machine [8]. The manual data recording requires sophisticated segmentation algorithm, which performs segmentation along the input or output spaces. The resultant set of segments consists of input vectors and corresponding vibration or friction patterns, which are used for the model building.

In [9] the two-dimesional input vector (normal force and velocity magnitude) of each acceleration pattern of vibrations denoted the position inside the look-up table, where the model output is approximated by using bilinear interpolation of four nearest model points. The acceleration pattern of each node is encoded into linear predictive coding (LPC) coefficients. This approximation model was improved in [3], by encoding acceleration patterns into auto-regressive moving average (ARMA) coefficients and by using Delaunay triangulation for model points interpolation. An alternative solution was proposed by Shin et al., where a frequency decomposed neural network was used instead of the look-up table, with similar input and output [8]. The most recent work is devoted to modeling and rendering of anisotropic haptic textures with three-dimensional model space [2], [10]. In this approach, the movement direction was implied by an additional two-dimensional velocity vector. The output acceleration patterns were encoded into line spectral frequency (LSF) coefficients.

All these data-driven haptic texture models have a common concept. The interpolation model gets the input vector and produces the time series output, meaning that a single input vector produces a continuous signal of the feedback pattern, i.e., vibration or variable friction. This similarity makes data-driven haptic texture models interchangeable. By relying on this commonality, this paper presents the cross-model sample selection algorithm.

Various sample selection algorithms were proposed for data-driven modeling of force responses during an object deformation. Sianov et al. proposed the method of spatial segmentation of the object into relatively homogeneous regions, where each region was represented by data of a single palpation [11]. In their other work, three algorithms were introduced for the selection of representative input-output data pairs [12]. The technique that provided the best approximation was based on k-d tree data structure, where barycenters of each leaf of the tree form the representative set of samples. The alternative approach was presented in [13], where the input space was partitioned in a way that the ratio of the standard deviation of corresponding output cluster to its mean remains below the perceptual threshold. Mean samples of each cluster are selected as representative. The main advantage of this approach is that it deals with a multidimensional output stimuli.

A wide range of sample selection algorithms are available in machine learning and data mining literature [14]. However, most of them are designed for classification and regression problems. The output of the model in classification problem is a class label. Meanwhile, the model output in regression problem is a continues value. In both cases, single input vector has a constant corresponding output value. On the other hand, the output of the data-driven haptic texture model is a feedback pattern, which is represented by a time series signal. Thus the available literature algorithms are unsuitable.

III. ALGORITHM

In this section, we propose a novel algorithm for representative sample selection across multiple recording trials. The main aim of this algorithm is to populate the input space by significant model points from multiple trials while reducing the number of outliers. Furthermore, a generic haptic texture model is also provided. This generic model provides the necessary platform to other haptic texture modeling algorithms to benefit from the aforementioned sample selection algorithm.

Despite the fact that none of available sample selection algorithms can be directly applied to model point selection for data-driven haptic texture modeling, the idea of several sample selection algorithms can be generalized and extended for this task. For example, Edited and Condensed Nearest Neighbor (ENN [15] and CNN [16]) were initially designed for classification task based on k-Nearest Neighbors (k-NN) classifier. The former algorithm is usually used for outlier reduction, whereas the later one eliminates redundant samples from the given set. Recently, Arnaiz-Gonzalez et al. adopted the idea of CNN and ENN for regression tasks [17].

Inspired by the work in [17], we extended the idea of ENN and CNN for the representative model point selection for data-driven haptic texture modeling. Instead of using the k-nn classifier, the general haptic texture model can be used in our approach. Before going into details about the proposed algorithm, we provide a brief introduction of the general haptic texture model.

A. General Haptic Texture Model

The evolution of the haptic texture models is provided in Sec. II. Even though all haptic texture models have their own

Algorithm 1 Sample Selection Algorithm

```
Input: $M = \{(x_1, y_1), \ldots (x_n, y_n)\}$, $k$, $l$, $\tau_1$, $\tau_2$
Output: $\widehat{M} \subseteq M$
     Removing outliers:
 1: $\rho \leftarrow getAverageSparsity(M)$
 2: for $i = k+1$ to $|M|$ do
 3:     $model \leftarrow train(M \setminus \{x_i, y_i\})$
 4:     $\widehat{y}_i \leftarrow model.simulate(x_i)$
 5:     $d \leftarrow getDistance(y_i, \widehat{y}_i)$
 6:     $\widehat{\rho} \leftarrow getLocalSparsity(m_i, M)$
 7:     $\theta \leftarrow \tau_1 + \alpha * (\widehat{\rho}/\rho - 1)$
 8:     if ($d > \theta$) then
 9:         $M \leftarrow M \setminus \{x_i, y_i\}$
10:     end if
11: end for
     Removing redundant patterns:
12: $\widehat{M} \leftarrow \{(x_1, y_1), \ldots (x_k, y_k)\}$
13: for $j = k+1$ to $|M|$ do
14:     $model \leftarrow train(\widehat{M} \cup \{x_j, y_j\})$
15:     $\widehat{y}_j \leftarrow model.simulate(x_j)$
16:     $d \leftarrow getDistance(y_j, \widehat{y}_j)$
17:     if ($d > \tau_2$) then
18:         $\widehat{M} \leftarrow \widehat{M} \cup \{x_j, y_j\}$
19:     end if
20: end for
21: return $\widehat{M}$
```

contributions, the conceptual representation of most models remain similar. The *model space* is an abstract coordinate system that describes the location of *model points*. Each model point is described by a location inside the model space and the feedback pattern $m_i = \{x_i, y_i\}$, where x_i denotes the $n-$ dimensional vector of the model points location, and y_i represents the feedback pattern $y_i = \{a_1, a_2, \ldots, a_n\}$. Thus the general haptic texture model can be described by the set of model points $M = \{m_1, m_1, \ldots, m_p\}$. An example is provided in Fig. 1. Since the data-driven model in most cases is an interpolation model, a given minimal set of model points M_{min} is required for a stable output. In some case the M_{min} consists of the synthetic model points, which marks the interpolation boundaries.

B. Proposed Algorithm

The pseudo code of the proposed method is depicted in Algorithm 1. The algorithm starts with the outlier reduction procedure (See lines 1 - 11), which is followed by redundant sample elimination (See lines 12 - 21). The input of the algorithm consists of an initial set of model points $M = \{\{x_1, y_1\}, \ldots \{x_n, y_n\}\}$, where the first k elements form the minimal set of model points. Threshold values τ_1 and τ_2 are used to control the reduction rate of outliers and redundant model points, respectively.

Outlier Reduction is an iterative process over the initial set M, where each model point $m_i = \{x_i, y_i\}$ is examined one at a time, starting from the $(k+1)^{th}$ element of the set. In each iteration, one model point is temporarily removed from the initial set $M \setminus (x_i, y_i)$. The resultant set is used for the model training. Following this, the feedback pattern $\widehat{y}$ is estimated by feeding the input vector x_i to the model. If the estimated $\widehat{y}_i$ and original y_i feedback patterns are considerably different, the probability that i^{th} sample is an outlier increases. This dissimilarity means that the contribution from the feedback pattern y_i contradicts to contributions of the neighboring ones. The dissimilarity between two feedback patterns is calculated by a dissimilarity metric, which is explained at the end of this section. The threshold value τ_1 denotes the level of dissimilarity, at which the model point is permanently removed from the set M.

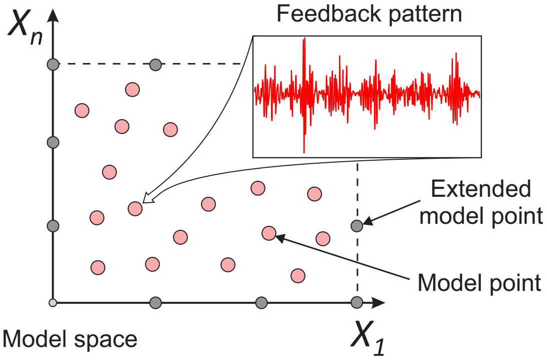

Fig. 1. General haptic texture model

This outlier detection strategy works well for dense regions, where the model point resembles to the neighboring ones. However, it can be misleading for sparse regions. The neighboring model points in sparse regions are usually different, since they are far from each other. Thus the threshold τ_1 should be adaptive to the local density of the model space. In order to solve this problem, the regularization term $\alpha * (\widehat{\rho}/\rho_i - 1)$ is introduced, where $\widehat{\rho}$ and ρ_i are average and local sparsity of the model space respectively. When the local sparsity equals to the average one, the regularization term turns to zero. Similarly, when the local sparsity is higher then the global one, the adaptive threshold value θ is increased, and the other way around. The parameter α represents the sensitivity of the algorithm to the local density. It is recommended to estimate the α by using the following equation.

$$\alpha = \tau_1 * \frac{\widehat{\sigma}}{\widehat{\rho}}, \tag{1}$$

where the $\widehat{\sigma}$ denotes the mean deviation of local sparsity at each model point from the average sparsity $\widehat{\rho}$. In order to estimate the local sparsity ρ_i of each model point m_i for a two-dimensional model space, we built the Delaunay triangulation by excluding the target model point m_i, and computed the average distance from m_i to three enclosing neighbors. Similarly, the average distance to four surrounding model points of the tetrahedron represented the local density for a three-dimensional model space.

Redundant Sample Elimination. Unlike the previous stage, the process of redundant sample elimination starts with the

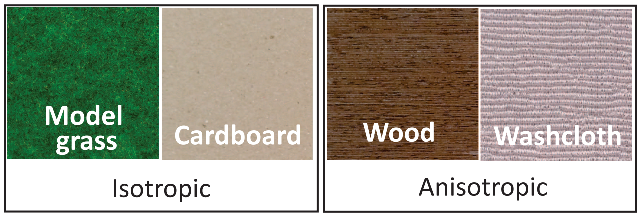

Fig. 2. Haptic Texture Datasets

minimal set, which contains only k model points. In every iteration, the haptic texture model is trained by using the set $\widehat{M}$. The set $\widehat{M}$ is extended by the candidate model point m_i, if the difference between the original and simulated feedback patterns exceeds the threshold τ_2. This iterative process finishes when all samples from M are assessed.

C. Error Metric

The error metric used for comparing the acceleration patterns is the spectral rms error. It is the difference between the approximated $\widehat{a}[n]$ and the recorded acceleration pattern $a[n]$. The equation for spectral rms error is given as:

$$E_n = e_n(\widehat{a}[n]) = \frac{RMS(F(\widehat{a}[n]) - F(a[n]))}{RMS(F(a[n]))}, \tag{2}$$

where RMS is the root mean square operator in the frequency domain, and $F(.)$ is the discrete Fourier transform. This error metric is preferred since it provides a better account of the perceptual differences as compared to the time domain error metrics.

IV. EXPERIMENTAL SETUP AND RESULTS

In this section, the performance evaluation of the proposed algorithm is provided. The algorithm performance can be considered as reliable, if it successfully achieves certain performance metrics. First, it should achieve a higher rendering quality after data reduction or at worst remain the same as earlier. Second, the number of outliers in the reduced data-set should be minimized. Last, the algorithm should be flexible enough to incorporate a multidimensional model space, i.e., the effect of increase in input dimensions should remain minimal.

In order to meet the above mentioned performance metrics, it was decided to use the data-driven model of anisotropic haptic textures [2] as an underlying model for the proposed sample selection algorithm. This model relates three-dimensional input (two-dimensional velocity vector, and normal force) with an acceleration pattern of vibrations generated from a tool-surface contact. Additionally, by replacing the three-dimensional input with a two dimensional one, this algorithm can readily be used to represent the isotropic textures as well.

A. Recording Setup and Dataset

The data from a tool-surface interaction are recorded by using a custom recording setup, which is illustrated on Fig. 3. The tool-surface interaction can be described by the position of the tool-tip, normal force and the vibration response.

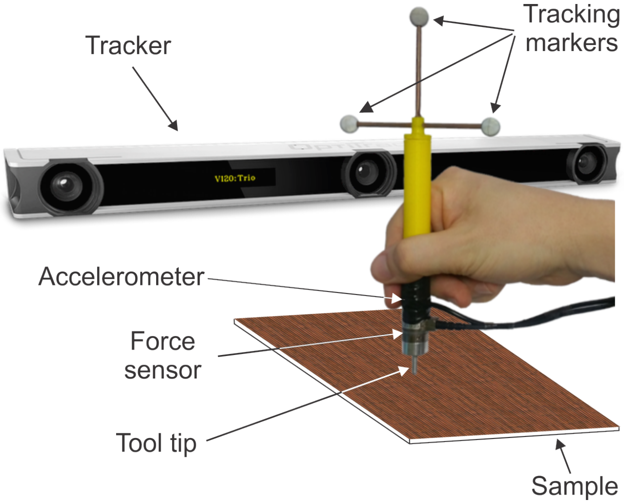

Fig. 3. Recording setup

The tool-tip position is tracked based on the position and orientation of three markers using an optical tracking device (V120:Trio; OptiTrack). The normal force is estimated based on the tool orientation and three dimensional force signal, which is acquired from a force/torque sensor (Nano17; ATI Technologies). Contact vibrations are recorded by a three-axis accelerometer (ADXL335; Analog Devices), where the acceleration signal along the normal direction of the sample surface was used for an experiment. The force/torque sensor and accelerometer are connected to an NI DAQ acquisition board (PCI-6323; National Instrument). A desktop PC with i7-6700K CPU and 32 Gb DDR-4 RAM was used for data collection and processing.

In order to evaluate the proposed algorithm, a dataset containing two isotropic and two anisotropic haptic textures was selected (see Fig. 2). Isotropic textures were scanned 20 times, where the duration of each scan was 20 seconds. The number of acquisition trials for anisotropic haptic texture was 40, whereas the trial duration was 20 seconds. The diameter of the tool tip used for recording was 3 mm.

B. Model Point Extraction

As a first step, the set of model points and corresponding acceleration patterns are extracted from the raw data. The raw signals from every recording trial are concatenated into a single data sequence. The position and normal force signals are low-pass filtered by using a Bessel filter with a cutoff level of 20 Hz. The acceleration signal was band-pass filtered with the bandwidth limits set at 10 and 1000 Hz. The lower limit filters out the gravity component, whereas the upper limit removes the noise.

A two-dimensional velocity signal is derived from the position data. The velocity signal and normal force are normalized from 0 to 1 and are used as an input signal for the anisotropic haptic texture model. In case of isotropic haptic textures, the two-dimensional velocity signal is replaced by the velocity magnitude , which is in turn normalized in the same way.

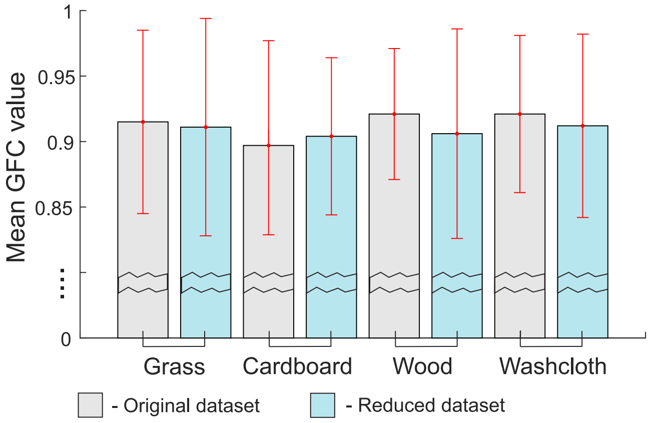

Fig. 4. Mean Goodness-of-Fit Criterion (GFC) for comparison between the original and reduced datasets. The error bars represent the standard deviation of GFC values.

In order to extract model points and corresponding vibration patterns, the input-space-based segmentation algorithm from [2] was used. This sample selection algorithm requires a minimal set of samples that guarantee a stable output during the selection process. Therefore, it was decided to compute convex hulls for each dataset and extract the boundary model points of the convex hull to form the above mentioned minimal set. This ensures that the sample selection process will be performed inside the convex hulls.

Since the proposed algorithm is iterative in nature, it is recommended to randomize the order of the model points before sample reduction. However, in the current study, it was empirically established that the data points are already scattered in a random fashion and further randomization was not required.

C. *Outlier Generation*

The performance of the outlier reduction algorithm can be measured by the detection rate. The detection rate is the ratio of the number of detected outliers to the total number of outliers. However, the detection of outliers within a given set requires a high level of deliberation. One possible solution can be to assume that the given set of model points is free of any outliers and manually include a synthetic set of outliers to it.

The synthetic set of outliers can be generated within the given set of model points by adding a noise signal to the corresponding vibration signal. However, the addition of simple colored noise can be unrealistic and prone to manipulation.

Another possible solution is to simulate the noise that causes the vibration signal distortion during the data collection process. The data from the object surface is usually collected on a table. In this case, mechanical noises propagate through the table and are recorded along with the vibration responses from the tool-surface interaction. It is next to impossible to filter out such noises from the vibration signal, since both the vibration and noise signals share a similar frequency band.

TABLE I
HAPTIC TEXTURE DATASETS

	Initial set	Minimal set	Outliers	Texture type
Grass	236	15	23	isotropic
Cardboard	253	18	25	isotropic
Wood	432	72	43	anisotropic
Washcloth	461	68	46	anisotropic

TABLE II
SELECTION RESULTS

	Detected outliers	Compression rate	Performance gain
Grass	18 / 23 = 0.78	236 / 87 = 2.71	-0.0059
Cardboard	20 / 25 = 0.80	253 / 104 = 2.43	0.0078
Wood	32 / 43 = 0.74	432 / 202 = 2.13	-0.051
Washcloth	32 / 46 = 0.69	461 / 181 = 2.54	-0.0092

A wide range of mechanical noises can be encountered during the data collection process. The noise signals used in this study were collected from the sudden collision of third-party objects with a table in home, office, or library environment. Six such noise signals were extracted from the dataset provided in [18]. The duration of each noise signal was 2 seconds. These noise signals were down-sampled to a 1000 Hz sampling rate and high-pass filtered with a 10 Hz cutoff frequency. Afterwards, we selected 10% of the model points from each dataset and added noise to its corresponding vibration pattern. In order to make the experiment fair, the noise signal was normalized using the following equation.

$$A_i' = A_i + N_i * \frac{\sigma_a}{\sigma_n} * \sqrt{\theta}, \qquad (3)$$

where θ denotes the target ratio of the variance of noise signal N_i to the variance of the vibration pattern A_t. σ_n and σ_a represent variance of the noise signal and vibration pattern, respectively. In other words, equation 3 normalizes the noise signal in such a way that the resultant variance of the noise is θ times the variance of the vibration pattern. The value of θ for this experiment was set at 0.4. This value is considered optimal, since it allows A_i to remain the main contributor to the resultant signal A_i', whereas the noise content is also maintained at such a level that it remains perceptually discriminable.

V. EVALUATION

In this section, the proposed sample selection algorithm was evaluated using the four datasets (See Table I) discussed in the previous sections. In order for the sample selection algorithm to perform successfuly, reasonable threshold values must be selected. These threshold values (τ_1 and τ_2) were selected in accordance with studies reported in [19], [8]. They evaluated their algorithms using the same error metric, i.e., relative spectral RMS. According to [19], [9] and later validated by [8], it is reasonable to consider an error in the range 0.29 - 0.42 as perceptually insignificant. Therefore, the value of τ_1 and τ_2 was set at 0.25 and 0.42, respectively.

The proposed algorithm was executed using the thresholds mentioned above. The performance of the algorithm was measured using several metrics. First, the rate of outlier detection was calculated, i.e., the number of true outliers detected as compared to the total number of outliers. Second, the compression rate of the algorithm was calculated, i.e., the reduction in number of data points after sample selection. The results for these two metrics are provided in Table II. It can be seen that the algorithm shows a high outlier detection rate for all the datasets. Furthermore, it also provides atleast a two fold compression rate across all the datasets.

Since, the ground truth data of the redundant subset of model points is not available, it is impossible to calculate the confusion matrix for the proposed algorithm. However, in order to evaluate if the proposed algorithm has removed significant model points, we compared the approximation performance of the original and the reduced datasets. For this purpose, new data was collected from tool-surface interaction for all datasets. The interaction time for this data was 10 seconds. The output vibration patterns from this data were considered as ground truth for evaluating the two point models. While, the input data was used as a test input for the algorithms. The approximated vibration patterns from the two point models were compared with the ground truth vibration patterns. This comparison was carried out using a Hernandez-Andres Goodness-of-Fit Criterion (GFC) to calculate the error between the power spectrums. According to this metric, a value of zero means no linear relationship, while a value of one means perfect reconstruction. Furthermore, the authors in [20] provide that a GFC value greater than 0.90 is considered as a good match. The procedure provided in [2] is followed to compute the mean GFC values. We refere the reader to [2] for the detailed description.

The mean GFC values for both the point models are provided in Fig. 4. It can be seen that the point models show almost the same level of accuracy. The relative difference of the two models is called as performance gain, showed in Table II. Here, a negative performance gain indicates that the reduced point model provided an accuracy value lower than the original point model.

VI. CONCLUSIONS AND FUTURE WORKS

In this study, a novel algorithm for representative sample selection across multiple tool-surface interaction recording trials was proposed. This algorithm can pave the way for seemlessly increasing the dimensionality of input model space for data-driven haptic texture rendering. Furthermore, it can be readily used with any underlying haptic modeling algorithm, without compromising on the rendering quality.

In the current algorithm, the outlier detection is carried out iteratively. In certain cases the iterative nature of detection can lead to erroneous results i.e., it can introduce an ordering bias where a true outlier remains in the data due to the earlier wrong detection of a data point. As a possible avenue for future research, a robust technique needs to be developed to ensure against any such ordering bias. Additionally, the outlier detection rate of the algorithm should be higher as compared to the current state to provide an even higher rendering quality.

ACKNOWLEDGMENTS

This research was supported by Global Frontier Program through NRF of Korea (NRF-2012M3A6A3056074) and by ERC program through NRF of Korea (2011-0030075).

REFERENCES

[1] M. Minsky, O.-y. Ming, O. Steele, F. P. Brooks Jr, and M. Behensky, "Feeling and seeing: issues in force display," in *ACM SIGGRAPH Computer Graphics*, vol. 24, no. 2, 1990, pp. 235–241.

[2] A. Abdulali and S. Jeon, "Data-driven modeling of anisotropic haptic textures: Data segmentation and interpolation," in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, 2016, pp. 228–239.

[3] H. Culbertson, J. Unwin, and K. J. Kuchenbecker, "Modeling and rendering realistic textures from unconstrained tool-surface interactions," *IEEE Transactions on haptics*, vol. 7, no. 3, pp. 381–393, 2014.

[4] W. B. Messaoud, M.-A. Bueno, and B. Lemaire-Semail, "Textile fabrics texture: From multi-level feature extraction to tactile simulation," in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 2016, pp. 294–303.

[5] D. J. Meyer, M. A. Peshkin, and J. E. Colgate, "Tactile paintbrush: A procedural method for generating spatial haptic texture," in *2016 IEEE Haptics Symposium (HAPTICS)*. IEEE, 2016, pp. 259–264.

[6] C. Basdogan, C. Ho, and M. A. Srinivasan, "A raybased haptic rendering technique for displaying shape and texture of 3d objects in virtual environments," in *ASME Winter Annual Meeting*, vol. 61, 1997, pp. 77–84.

[7] J. P. Fritz and K. E. Barner, "Stochastic models for haptic texture," in *Photonics East'96*. International Society for Optics and Photonics, 1996, pp. 34–44.

[8] S. Shin, R. H. Osgouei, K.-D. Kim, and S. Choi, "Data-driven modeling of isotropic haptic textures using frequency-decomposed neural networks," in *World Haptics Conference (WHC), 2015 IEEE*. IEEE, 2015, pp. 131–138.

[9] J. M. Romano and K. J. Kuchenbecker, "Creating realistic virtual textures from contact acceleration data," *IEEE Transactions on Haptics*, vol. 5, no. 2, pp. 109–119, 2012.

[10] A. Abdulali and S. Jeon, "Data-driven rendering of anisotropic haptic textures," in *Proceedings of the Asia Haptics 2016 Conference*, December 2016.

[11] A. Sianov and M. Harders, "Data-driven haptics: Addressing inhomogeneities and computational formulation," in *World Haptics Conference (WHC), 2013*. IEEE, 2013, pp. 301–306.

[12] R. Höver, M. D. Luca, and M. Harders, "User-based evaluation of data-driven haptic rendering," *ACM Transactions on Applied Perception (TAP)*, vol. 8, no. 1, p. 7, 2010.

[13] A. Abdulali, W. Hassan, and S. Jeon, "Stimuli-magnitude-adaptive sample selection for data-driven haptic modeling," *Entropy*, vol. 18, no. 6, p. 222, 2016.

[14] I. Triguero, J. Derrac, S. Garcia, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 86–100, 2012.

[15] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 408–421, 1972.

[16] P. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, pp. 515–516, 1968.

[17] Á. Arnaiz-González, M. Blachnik, M. Kordos, and C. García-Osorio, "Fusion of instance selection methods in regression tasks," *Information Fusion*, vol. 30, pp. 69–79, 2016.

[18] A. Mesaros, T. Heittola, and T. Virtanen, "Tut acoustic scenes 2016, evaluation dataset," Nov. 2016. [Online]. Available: https://doi.org/10.5281/zenodo.165995

[19] J. M. Romano, T. Yoshioka, and K. J. Kuchenbecker, "Automatic filter design for synthesis of haptic textures from recorded acceleration data," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1815–1821.

[20] H. Culbertson, J. M. Romano, P. Castillo, M. Mintz, and K. J. Kuchenbecker, "Refined methods for creating realistic haptic virtual textures from tool-mediated contact acceleration data," in *Haptics Symposium (HAPTICS), 2012 IEEE*. IEEE, 2012, pp. 385–391.