Data-Driven Rendering of Anisotropic Haptic Textures

Arsen Abdulali and Seokhee Jeon

Department of Computer Science and Engineering, Kyung Hee University,
Yongin-si 446-701, Republic of Korea
{abdulali, jeon}@khu.ac.kr

Abstract. It is common to interact with a wide range of surfaces with anisotropic haptic textures in our daily life. However, the rendering techniques were not capable of incorporating the directional grain in a virtual world. In our previous work, we have proposed a data-driven model for anisotropic haptic textures, which stores and interpolates contact vibration patterns. In this paper, we have developed a complementary rendering algorithm. This algorithm has been implemented in form of a cross-platform computing library, and deployed to the tablet-PC-based demonstration setup. A set of eight anisotropic and one isotropic textures is prepared for the demonstration. Two demonstration scenarios have been provided for the realism evaluation.

Keywords: Data-Driven Rendering, Anisotropic texture, RBF Networks

1 Introduction and Background

Nowadays, the virtual and real worlds are on the verge of being undistinguishable by human. The immersion into virtual reality is due to high fidelity displays and sophisticated rendering algorithms. Modern rendering algorithms are interactive meaning that they produce feedback accordingly to the physical input of the user. In order to produce a realistic feedback of a touch, the rendering algorithm should consider not only object properties as an input, but the physical input of a user as well.

One of the most complex rendering techniques is haptic texture rendering. The interaction with a surface texture happens either bare-handed [5] or using a tool [2]. In both cases a perceived response changes accordingly to the physical input, such as a stroking velocity and direction, a pushing force, etc.

The output response in modern haptic texture rendering algorithms is decided based on the model. The model can be physics based, where the physical process of the touch is simulated, or data-driven, where the model is built based on measurements of real interactions. One of the most successful data-driven model of the haptic texture is evolved in a research group lead by K. Kuchenbecker [2, 3]. Their model maps the normal force and velocity magnitude of the contact with vibration patterns that are propagated through the tool during tool-surface interaction. These vibration patterns are encoded in autoregressive (AR) coefficients. The rendering application of this model is available in their recent work [3], where they also introduced a library with one-hundered virtual object surfaces. Since this model is limited to homogenies istropic textures, a wide renge of anisotropic surfaces could not be modeled. Recently, an alternative solution was proposed for modeling of isotropic haptic textures in [6]. Instead of using AR model interpolation, the vibration patterns are stored inside a frequency-decomposed neural network model. This model can be extended to anisotropic texture rendering. However, it is less applicable, since it requires special data collection equipment for the model building. In our recent work [1], we proposed a new approach for modeling anisotropic haptic textures. This method includes two main contributions that makes it possible to build a model of anisotropic haptic texture. First, a new input-space-based segmentation algorithm is developed, which produces stationary acceleration segments with corresponding nearly constant input. Second, a flexible RBFN based model for AR acceleration patterns storage and interpolation is introduced, which makes the increase of the input dimension computationally inexpensive.

In this paper we present a complementary rendering algorithm of anisotropic haptic texture models. This algorithm is developed in form of a cross-platform computing library. Additionally, we built a set of eight anisotropic and one isotropic haptic texture models for performance demonstration of the computing library. We also prepared two demonstration scenarios that will help participants to evaluate the realism of virtual haptic textures.

2 Modeling Procedure

In this section we will provide a brief explanation about the modeling algorithm to define the input of the rendering algorithm. For the complete information about the modeling procedure refer to [1].

2.1 Algorithm Outline

The data-driven model of anisotropic haptic texture maps a two-dimensional velocity vector and normal force of the tool-surface contact with corresponding vibration patterns that are propagated through the tool. This model is estimated based upon measurements of the real tool-surface contact.

The modeling algorithm of anisotropic haptic textures consists of three stages: *data acquisition and preprocessing*; *data segmentation*; and *model building*.

Data Acquisition and Preprocessing. In this stage, the contact data is acquired during unconstrained tool-surface interaction. The two-dimensional velocity vector is calculated based on the data from a position sensing device. The normal force is captured using a force sensor. The contact vibrations are captured by an accelerometer. The two-dimensional velocity vector and normal force are re-sampled to the same level of sampling frequency 1000 Hz and low-pass filtered to remove a noise. The acceleration signal is band-pass filtered in a range of 10-1000 Hz to remove a gravity component and noise.

Data Segmentation. The segmentation algorithm partitions the input and output signals into stationary acceleration patterns of contact vibrations having a corresponding nearly constant input. It is assumed that the contact vibration remains stationary if the movement trajectory of the tool remains in a straight line with a nearly constant velocity magnitude and normal force. Hence, the segmentation is performed along the input signals.

The segmentation algorithm is based on a bottom-up technique and consists of two stages. First, the position data is segmented into straight lines, where the line is considered straight if the maximal deviation of position points are below the threshold τ_1. Similarly, these segments are sub-segmented into signals having a nearly constant velocity magnitude and normal force, where the ratio of the standard deviation to the mean of the velocity magnitude and normal force is limited by the threshold τ_2.

Model Building. The contact vibration patterns (model output) and corresponding input vectors $\mathbf{u} =< v_x, v_y, f_n >$ are provided by segmentation algorithm for a model building, where v_x and v_y are mean values of two velocity components, and f_n is an average normal force of each segment.

The contact vibration patterns are converted to autoregressive (AR) models with a common order. The AR model describes time-varying process in such a way, where the output value depends linearly on its previous outputs. The AR model of each acceleration pattern is represented by m coefficients and the variance, where m is an order of the AR model. Due to the stability issue of the AR model interpolation [4], the AR coefficients are converted to corresponding line spectral frequency (LSF) ones.

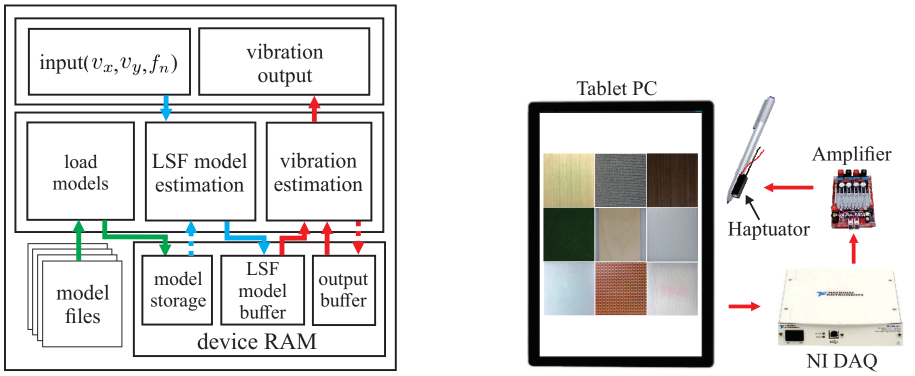
Finally, the RBFN model is trained using the set of input vectors and corresponding set of output acceleration patterns, where the output acceleration patterns are represented by LSF coefficients and the variance.

3 Rendering Setup

In this section we will describe a demonstration setup for anisotropic texture rendering algorithm. The setup consist of software and hardware components. The software component is implemented in form of a computing library to make it independent from the hardware. The software architecture of the computing library is depicted in Fig. 1(a). The hardware setup that will be used for demonstration is shown in Fig. 1(b).

3.1 Software Architecture

The architecture of the rendering software consists of three layers (See Fig. 1(a)). The upper layer is referred to as interactive layer. This layer computes the input vector $\mathbf{u}$ based on readings from the input device. Additionally, this layer displays response vibrations back to the user. The business logic of the anisotropic haptic texture library is represented by the second layer. This layer is developed in form



(a) Software architecture of the computing library.

(b) Hardware setup for algorithm demonstration.

Fig. 1. Hardware for recording and sample set.

of a platform independent computing library. The computing library consist of three functional blocks. First block loads the set of haptic texture models into device memory. The second block estimates the LSF (line spectrum frequency) coefficients and the variance by feeding the input vector $\mathbf{u}$ to the RBFN haptic texture model (See Sec. 2.1). The LSF coefficients and the variance are updated inside the buffer in 100 Hz. Meanwhile, the output vibration signal is generated by the third block, which runs on the other computing thread having frequency 2 kHz. The output vibration signal is produced based on buffered LSF coefficients, the variance and m buffered vibration outputs, where m is a number of LSF coefficients. Note also that all functional blocks work in separate computing threads. The frequency of each computing thread can be reset in accordance with user needs.

3.2 Vibration Estimation

The set of LSF coefficients and the variance describe the contact vibration pattern for a given vector $\mathbf{u}$ inside the input space of the RBFN haptic texture model. Therefore, the main task of the RBFN haptic texture model is to provide the mapping of three-dimensional input vector $\mathbf{u}$ with corresponding $(m+1)$-dimensional output vector (m LSF coefficients and the variance). This output vector can be calculated using following equation

$$f_i(u) = \sum_{j=1}^{N} w_{ij}\phi(\|u - q_j\|) + \sum_{k=1}^{L} d_{ik} g_{ik}(u), \qquad \mathbf{u} \in \mathbf{R}^n \tag{1}$$

where $i = \{1, ..., m+1\}$ denotes the index of LSF coefficients and the variance, w_{ij} is a weight constant and q_j is a center of the radial basis function. The functions $g_k(u)$ $(k = 1, ..., L)$ form a basis of the space $\mathbf{P}_p^n$ of polynomials with degree at most p of n variables.

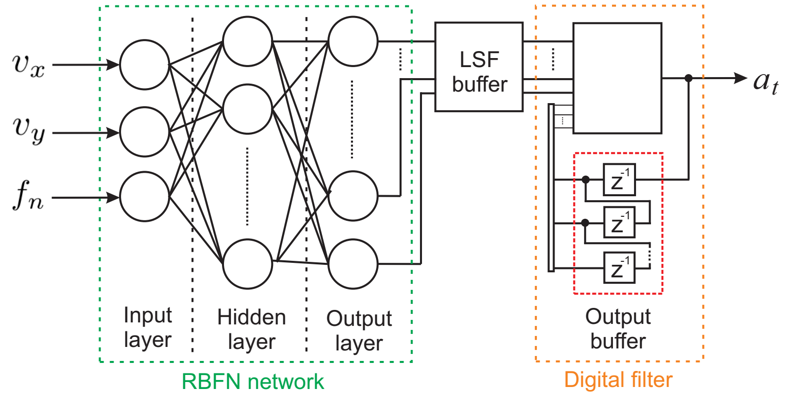


Fig. 2. Model storage and interpolation RBFN architecture.

The output vibration values are calculated using an approach similar to [2]. First, the LSF coefficients are converted to AR ones. Second, the AR coefficients, variance, and m buffered outputs are fed to the transfer function of the Direct Form II digital filter

$$H(z) = \frac{\varepsilon_t}{1 - \sum_{k=1}^{p} w_k z^{-k}}, \tag{2}$$

where w_k are AR coefficients, ε_t is a random sample from a normal distribution. The output value of the transfer function (See Eq. 2) is the output acceleration value.

3.3 Hardware Setup and Implementation

In order to demonstrate the quality of the modeling and rendering algorithms, we designed a tablet-PC-based hardware setup (See Fig. 1(b)). The tablet PC (Surface Pro 4; Microsoft) was selected as a rendering device. The contact velocity is calculated based on the contact position data from the touch screen of the tablet PC. The normal force of the contact is calculated based on readings from active digital pen (Surface Pen; Microsoft) with a sensing capability of 1024 pressure levels. The output vibrations are displayed using NI DAQ data acquisition device (USB-6251; National Instruments). This output signal is amplified by an analogue amplifier and is displayed using a voice coil actuator (Haptuator Mark II; Tactile Labs).

The set of nine haptic texture models (See Fig.1(b)) was built and stored into model files, which are competible with the rendering application. The same set of samples had been evaluated in a spectral domain in our previous work [1]. This sample set consists of three groups of materials: hard plastic (S1-S3), wood (S7-S9), and fabric (S4-S6). Eight samples have anisotropic texture. The only isotropic sample is S4.

3.4 Demonstration Protocol

The audience will have a choice of two demonstration scenarios.

The first demonstration scenario is a simple exploration of virtual and real surfaces. The participant will see a grid of virtual samples on the screen of the tablet PC and a set of corresponding real samples. After exploration of each real-virtual pair, the participant is asked to give a feedback about the realism of rendered textures.

The second demonstration scenario is a matching test of rendered samples. The participant explores a grid of virtual samples, where image textures are hidden. The participant is asked to find a corresponding virtual pair for each real sample. When the participant finishes this task, image textures of samples are displayed and the number of matching pairs can be counted. Similarly, the participant is asked to give a feedback about the realism of rendered textures.

4 Conclusion

In this work we developed a computing library for anisotropic haptic texture rendering. This computing library is deployed on a table-PC-based hardware setup. In order to show the rendering quality to the audience, the set of nine virtual haptic texture models was built and stored into model files. Two demonstration scenario are proposed, which will reveal the realism of the virtual haptic textures.

References

1. Abdulali, A., Jeon, S.: Data-driven modeling of anisotropic haptic textures: Data segmentation and interpolation. In: International Conference on Human Haptic Sensing and Touch Enabled Computer Applications. pp. 228–239. Springer (2016)
2. Culbertson, H., Unwin, J., Kuchenbecker, K.: Modeling and rendering realistic textures from unconstrained tool-surface interactions. Haptics, IEEE Transactions on 7(3), 381–393 (July 2014)
3. Culbertson, H., Delgado, J.J.L., Kuchenbecker, K.J.: One hundred data-driven haptic texture models and open-source methods for rendering on 3d objects. In: 2014 IEEE Haptics Symposium (HAPTICS). pp. 319–325. IEEE (2014)
4. Erkelens, J.S.: Autoregressive modeling for speech coding: Estimation, interpolation and quantization. Chapter 4: Spectral Interpolation. TU Delft, Delft University of Technology (1996)
5. Saga, S., Deguchi, K.: Lateral-force-based 2.5-dimensional tactile display for touch screen. In: 2012 IEEE Haptics Symposium (HAPTICS). pp. 15–22. IEEE (2012)
6. Shin, S., Osgouei, R., Kim, K.D., Choi, S.: Data-driven modeling of isotropic haptic textures using frequency-decomposed neural networks. In: World Haptics Conference (WHC), 2015 IEEE. pp. 131–138 (June 2015)