DOI: 10.1109/ICRA48891.2023.10160503

Model-Mediated Teleoperation for Remote Haptic Texture Sharing: Initial Study of Online Texture Modeling and Rendering

Mudassir Ibrahim Awan[1]*, Tatyana Ogay[1]*, Waseem Hassan[1]*, Dongbeom Ko[2], Sungjoo Kang[2], Seokhee Jeon[1]

***Abstract*— While model-mediated teleoperation (MMT) is an effective alternative for ensuring both transparency and stability, its potential in transmitting surface haptic texture is not yet explored. This paper introduces the first MMT framework capable of sharing surface haptic texture. The follower side collects physical signals contributing to haptic texture perception, e.g., high frequency acceleration, and streams them to the leader side. The leader side uses the signals to build and update a local measurement-based texture simulation model that reflects the remote surface. At the same time, the leader runs local simulation using the model, resulting in non-delayed, stable, and accurate feedback of texture. Considering that rendering haptic texture needs tougher real-time requirements, e.g., higher update rate and lower action-feedback latency, MMT can be a perfect platform for remote texture sharing. An initial proof-of-concept system supporting single and homogeneous surface is implemented and evaluated, demonstrating the potential of the approach.**

I. INTRODUCTION

In recent years, teleoperation systems in general have received significant attention in an effort to develop sophisticated and robust applications across various walks of life, e.g., robotic surgery [1], nuclear waste management [2], remote environment exploration [3], etc. These systems also employ different sensors and actuators to enable physical interaction (haptics) and relay the physical properties of the remote/follower side to the leader side. Unlike visual or aural feedback, haptic feedback requires a significantly higher update rate and extremely lower latency to maintain realism, whereas network systems for teleoperation are inherently unreliable in bandwidth and latency [4], [5].

Various techniques have been used to overcome the limitations in haptic teleoperation, e.g., different versions of wave variable transformation [6] and time domain passivity approach [7]. However, these systems run a trade-off between system transparency and stability. One of the technologies that can effectively bypass this trade-off is the Model-Mediated Teleoperation (MMT), which, in an ideal case, can provide a stable and time-delay-invariant teleoperation [8], [9].

MMT has recently been attracting attention. In a standard model of MMT, a leader side keeps a local simulation model of a remote environment, which is used to simulate and generate non-delayed haptic feedback in real-time. The core enabling technology is the online construction, update, and simulation of the local model. In a majority of MMT systems, the follower side already estimates the parameters for geometric and contact dynamic properties of the remote environment. Once estimated, the follower sends only these parameters to the leader side, instead of sending the whole interaction and sensed data. The leader side utilizes the parameters to construct the local model and keep it updated [9], [10], [11].

A majority of haptic teleoperation systems including MMT systems have mainly focused on kinesthetic feedback, especially force feedback [12], [13], as it is one of the most salient features of haptic feedback[14]. Some teleoperation systems have been developed to provide tactile feedback, such as, shape/geometry [15], friction [16], thermal rendering [17], etc. Several studies have provided both the kinesthetic and tactile feedback in an effort to deliver higher realism [18].

However, to the best of our knowledge, haptic texture, an important haptic property of materials, seems to have gone largely unnoticed in teleoperation and especially in the field of MMT. This is mainly due to the lack of proper modeling/simulation method for surface texture rendering with reasonable accuracy. However, the last 10 years saw emerging data-driven texture modeling and rendering methods that guarantee realism and real-time requirement [19], [20], [21], [22], [23], [24], [25], and timely application of the methods to MMT would open new possibility in teleoperation.

Haptic texture sharing can be a perfect candidate for MMT. In general, remote delivery of haptic texture is more sensitive to network performance due to the two requirements for accurate rendering of texture; 1. relatively higher update rate and 2. extremely low leader-follower latency. Haptic texture is usually manifested in the form of high frequency (up to 500 Hz) vibration, which requires an update rate of more than 1000 Hz for a seamless regeneration. Very low leader-follower transmission delay is needed due to the fact that action-response synchronization is critical for perception of texture. Very small but noticeable delay often completely destroys the perception of texture [26], [27], [28], e.g., stroking stopped but vibration remains, which eventually breaks the presence. Closed-loop teleoperation systems intrinsically suffer from variable time delays.

*These three authors have equally contributed. Seokhee Jeon is the corresponding author.

[1] Authors are with Department of Computer Science and Engineering, Kyung Hee University, South Korea {miawan,ogay,waseem.h,jeon}@khu.ac.kr

[2] Authors are with AI Research Laboratory, ETRI, South Korea {dbko112,sjkang}@etri.re.kr

In the current paper, we present a new bilateral model-mediated teleoperation framework that can deliver haptic texture. A follower robot interacts with the surface texture of a remote site and sends the interaction and response signals to the leader side. A local simulation model that represents the haptic texture of the remote surface is built and updated at the leader side in real-time using the data from the follower side. At the same time, rendering of the texture occurs at the leader by running the local model. The quality of rendering increases as the local model is updated with the signals coming from remote interaction.

II. Background

A. Acceleration-Based Haptic Texture Modeling

Texture of a surface is defined by a small-scale micro and macro geometry of the surface. In particular, perception of surface *haptic* texture when stroking a surface (with bare-hand or a rigid tool) is the result of complex contact dynamics between the geometry profile of the contacting surface and the tool/skin. The dynamics yields changes in various physical signals, i.e., small normal/lateral displacement/force/acceleration. These changes are perceived by the mechanoreceptors in the skin, generating perception of haptic texture.

There are two different approaches for digitizing the haptic texture. First and quite straightforward way is to model this micro/macro geometry itself [29]. The drawback of this method lies in rendering: it is not very straightforward to accurately simulate the contact dynamics in so-called haptic real-time, i.e., more than 1kHz update rate. Due to this, many simplifications are often introduced [30], [31]. Second approach avoids the complex dynamics simulation by just storing the resultant physical signals (and associated interaction parameters). The approach then simply interpolates the signals with regard to the current user's interaction for replicating the dynamics for rendering. This approach is particularly effective under tool-mediated interaction where physical signal due to stroking comes in the form of single vibration (or high frequency acceleration) of a rigid-body tool. Thus, modeling and rendering of texture can be reduced to how to parameterize, store, and regenerate these high frequency acceleration signals [19].

A series of works has recently been introduced for the second approach, as the name of data-driven texture modeling and rendering. Kuchenbecker et al. captured high frequency vibrations, from tool-surface interaction, in correspondence with input force and speed [32], [33], [34]. The parameterized acceleration profile samples are stored in the speed-force interaction space and then interpolated to synthesize the signals for rendering under a given arbitrary user interaction. The synthesized acceleration is turned into high frequency vibrations, generating the perception of texture. Abdulali et al. improved this idea by catering for the direction of interaction in addition to force [22]. This improvement enabled the rendering of anisotropic textures. Another study trained Generative Adversarial Networks (GAN) to produce high frequency vibrations from images of textures [35].

B. Model-Mediated Teleoperation

MMT is also known as impedance reflecting or virtual reality-based teleoperation. It was first introduced by Hannaford [8] and further improved by Niemeyer et al. [9], [36]. MMT is mainly designed to ensure transparent and stable teleoperation in the face of time varying communication delays or data loss.

MMT systems has three main challenges: 1) precise online modeling of the remote environment, 2) updating the estimated model in the face of arbitrary time delay, 3) local simulation of the model for rendering. In MMT, a virtual avatar of the remote site is modeled based on the interaction with remote objects. Instead of transmitting the complete signal at every point, estimated model parameters are transmitted to the leader side and a copy of the remote environment is established.

III. Overview of Our Approach

This paper aims to build an MMT framework for haptic texture. The Overall structure is illustrated in Fig 1. We assume that a surface at the remote site is palpated by a rigid end-effector of the follower robot via single-point contact. The interaction data of the end-effector, i.e., velocity and normal force (load against the surface), are measured and streamed to the leader server through the network. At the leader side, a local data-driven acceleration-based texture model is built and iteratively updated in real-time based on the streamed interaction and acceleration data. At the same time, the local texture model is simulated based on the current interaction parameters of the user at the leader side, providing non-delayed vibration-based texture feedback that reflects the remote surface. During the interaction, the local model is progressively improved. Different strategies can be used to cope with initial premature model, e.g., initially using direct replaying of the streamed acceleration, then switching to MMT when enough quality is guaranteed.

Different from standard MMT that only sends estimated parameters via the network, we decided to still stream the whole interaction and response data to the leader and to let the leader do the parameterization and model building. Two reasons are there. First, we thought the more critical limit is on the latency, not the bandwidth of the network, so we take advantage of having all necessary data at the leader side at the cost of bandwidth. Since the leader side has all the data, different methods can be applied depending on the network situation in hybrid a manner, e.g., traditional streaming-based feedback when network situation is good enough. In addition, parameterizing high frequency acceleration data needs complex computation, and a remote system usually has very limited computing power.

The data-driven acceleration-based texture modeling is commensurate with MMT. The approach is measurement-based, and MMT inherently has plentiful high quality (but delayed) data. It can inherently generate relatively high quality feedback with low computational power, which fit well the requirements of MMT. One of the drawbacks, low adjust-ability is not normally required in MMT where only

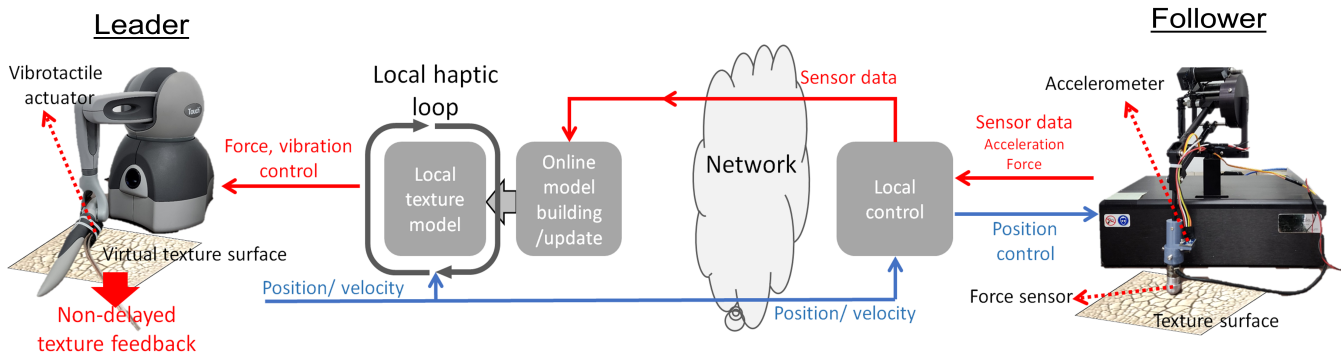

Fig. 1. The overall concept and software/hardware components implemented in this study.

exact copy is needed. Sample-interpolation-based rendering is suitable for progressive model building.

IV. Hardware and Data Transmission

The hardware components are presented in Fig. 1. Two haptic devices are used as the leader and follower robots. A Phantom premium (1.0; 3D systems) is used as the follower robot, while Touch X (3D systems) plays the role of the leader interface. A force sensor (nano-17;ATI Industrial Automation) and an accelerometer (ADXL335;Analog Devices) are attached to the end effector of the follower robot for data collection. A haptuator (MK2; Tactile Labs) is attached to the end-effector of the leader interface to render haptic vibration feedback.

A user interacts with the leader interface to explore the remote environment. The movements of the interface are replicated on the follower device that interacts with a textured surface. The force sensor and the accelerometer collect force and acceleration values from interaction, while the position data is collected from the follower robot's encoders. This data is transmitted over the network to train the local model on the leader side. The local model also receives position and force data from the leader interface, which is fed into the local model to render texture feedback at the haptuator attached to the interface. For kinesthetic feedback, the streamed force sensor reading is directly used for the leader interface to generate same force feedback.

The network framework used in this study is Socket.IO which provides real-time, event-based bidirectional communication between leader and follower environments. Two different edge servers communicate with each other over a shared WiFi network using Socket.IO. One edge server operates the leader device, while another edge server operates the follower device. In the current setup, the network delays vary between 1 ms and 5 ms.

V. Online Texture Modeling Framework

This section describes details about creating haptic texture models based on interaction signals with surfaces. The signals include force and speed of the end-effector as input, and high frequency acceleration (vibration) signals as output, originating from the interaction. The data collection, cleaning, and segmentation procedures are detailed first. Then the model formation and storage are briefly described.

For texture modeling, the overall approach is inspired by the acceleration-based texture modeling presented by Culbertson et al. [19]. The original system is designed for offline modeling. We revised all the necessary components so that they work in online with real-time streamed data, and so that the model update and simulation can be done simultaneously. The framework for MMT system along with online model update is provided in Fig. 2.

A. Data Processing and Segmentation

The data for modeling is generated when the follower robot interacts with a textured surface. The 3D position data (1 kHz) and force data (1 kHz) are used as inputs. The acceleration data (1 kHz) provided by the accelerometer is used as output. All the data is transmitted to the leader side where it is further processed.

Data segmentation is done at the leader server. In acceleration-based data-driven texture modeling, the output vibration for different input interaction parameters is sampled, stored, and then interpolated for rendering based on current user's interaction. To facilitate the interpolation, the vibration signal is parameterized using Auto-Regressive (AR) modeling (see next section). AR model needs nearly stationary input segments for proper parameterization. To this end, first, the signal is divided into straight lines where position is nearly constant. Second, the straight lines are further divided into segments carrying practically constant velocity and force magnitudes. For further details of the algorithm readers may refer to [23].

B. Model Creation and Storage

Once a new stationary segment is formed, a set of AR coefficients and variance value that reflect the segment is estimated. The process in this step is originated from [19]. The AR models for a single texture must contain the same number of coefficients, as these are used for interpolation during rendering. Afterwards, the AR coefficients are stored in the form of line spectral frequencies (LSF) for stability during rendering.

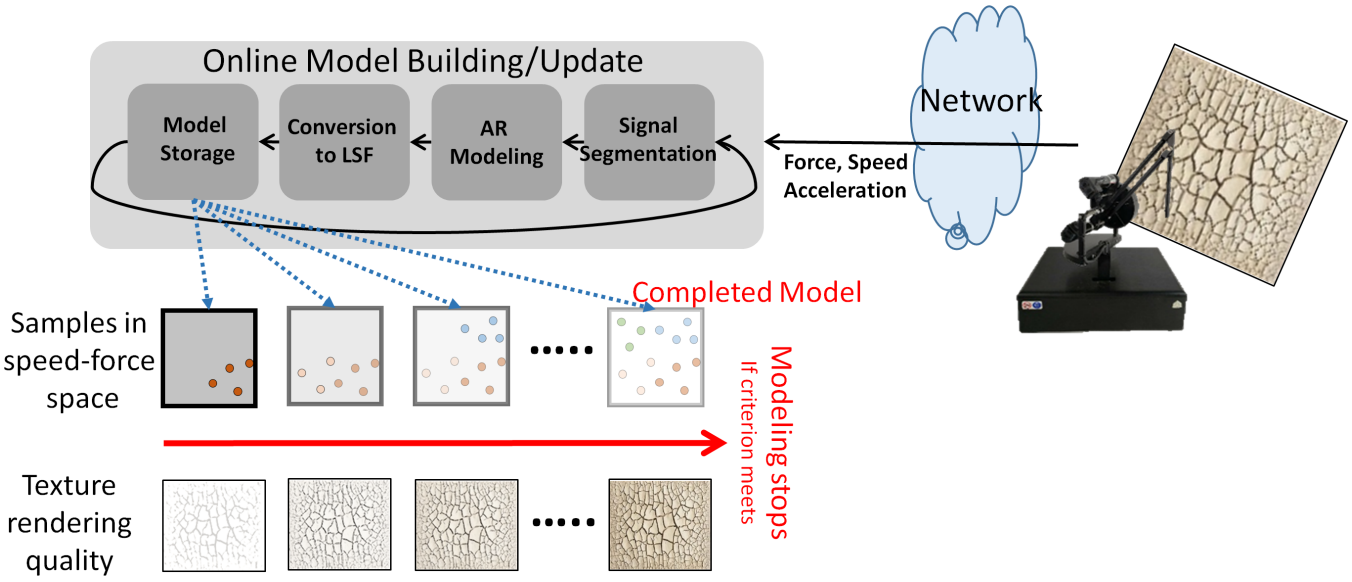

Fig. 2. The concept of iterative online model building and update. The interaction signals are streamed to the online modeling module. Four steps of online model update are iterative conducted, generating a progressively improving local model.

The AR models formed at the end are scattered in a 2D space where the axes are represented by speed and force. Delaunay triangulation is carried out in this scatter graph with normalized AR models, as shown in Fig. 3. The triangulation is extended to include the end points of the space. To this end, the max force and velocity are predefined in the current algorithm. After analyzing several AR models, it was decided to consider 300 mm/s as the maximum velocity, and 3 N as the maximum force. A total of five end points is added at each of the extremes (zero force and velocity, and maximum force and velocity) by copying the AR coefficients of their closest neighbors. The maximum end points also receive the variance of their closest neighbors, whereas those at the minimum receive zero variance.

C. Online Model Update

As an initial study, this paper assumes that interaction occurs with a single homogeneous and isotropic texture at one time. Future work will be followed for the management of multiple texture surfaces including the detection of encountering new surface, model switching, combining with geometry model and so on.

The model is updated in every one second. In second, data for last 1 second (1000 samples) is processed. 200 samples of the previous data are additionally appended at the beginning of the new one second data to ensure continuity. The new 1200 data are segmented and parameterized to AR models. New AR models are then inserted into the previously established speed-force sample database. This process continues as new data becomes available. Note that the update of the model is irrespective of the rendering output, i.e., rendering continues unhindered while the model is being updated.

The number of AR samples increase in the local model as new data becomes available. If the interaction occurs for a significant amount of time, a large number of redundant samples would appear in the model, which increases searching time and model size. In order to limit the number of samples to a manageable level and to ensure the compactness of the model, the system keeps track of how complete the model is and stops adding new samples when a criterion is met. To define the criterion, the speed-force space is divided into nine equal rectangles. The center point of each rectangle is defined, and the sum of distances from the three nearest AR models to the center point is calculated. As the model is updated, the total distance is gradually reduced. Once the sum of all distances reaches below a predefined threshold given in 1, model update is stopped. At this point the model is assumed to be good enough as it shows that there are enough samples in all regions of the input space.

The completion criterion is given by the following equation:

$$\hat{Y} = \sum_{1}^{j} \frac{N\sqrt{\alpha^2 + \beta^2}}{3}, \tag{1}$$

where α and β are the diagonal lengths of a rectangle, N is the number of nearest neighbors, and j is the total number of rectangles.

Note that at the beginning of interaction with a new surface, the local model is yet to be created or matured, and the system cannot generate accurate texture feedback. Our strategy to deal with this is to rely on the traditional method; direct application of streamed acceleration to the actuator. Once the system determines that the samples of the model is reasonably collected, the feedback generation is switched to the local model-based. The determination criterion is also dictated by a modified version of the completion criterion 1. From our intensive observations, a value of 7.5 ensures enough number of segments (on average

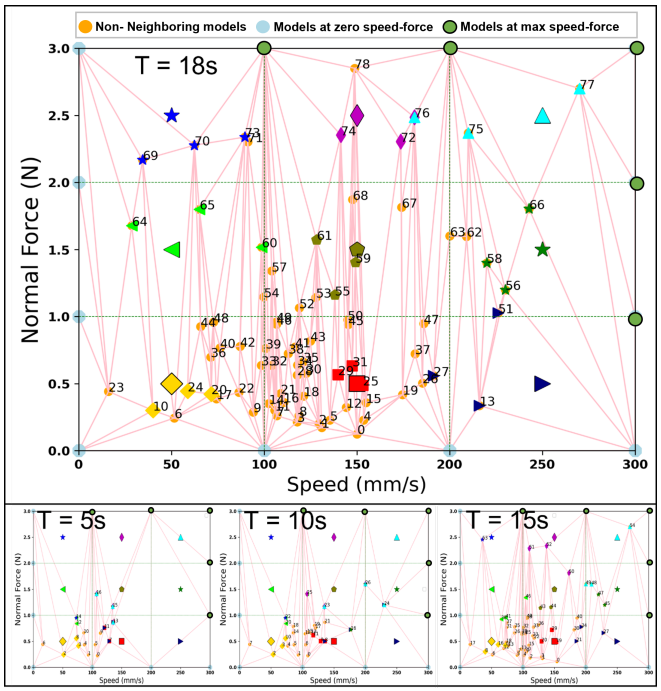

Fig. 3. Above: Actual completed model for Texture-S3 in speed-force space. Below three: progressively improving local models as time elapses. Red lines represents the result of Delaunay triangulation. Green lines are for the model completion testing. They divides the overall speed-force space into nine equal rectangles, each of which has a centroid from which the distances to the nearest samples are measured. Centroids is marked in the same color as its three neighboring AR models.

10 segments, achieved in roughly three seconds) that are reasonably well scattered across the speed-force space.

VI. Texture Rendering

This section describes the texture rendering framework employed at the leader side. Rendering occurs in a local loop at the leader side without delay. The rendering system used in this paper follows the algorithm provided in [19].

When a user interacts with the leader robot, the interaction force and speed values are captured. They are used to determine AR model for generating the vibration signal. Instead of simply selecting nearest sample in the speed-force space, interpolation scheme is used, which ensure smoother transition between samples. Direct interpolation of AR coefficients could potentially result in shifting the poles of the transfer function and make the whole model unstable [37]. Therefore, the AR models are first converted into LSF and interpolation is carried out among the LSF coefficients.

After the conversion, Delaunay triangulation is carried out in the AR space that encompasses all of the models that were created, along with those situated at the maximum and zero axis. The captured speed and force determine three nearest samples that constructs a triangle. Barycentric coordinates are used to generate new interpolated AR coefficients and variance values for the current point. As mentioned in Sect. V, the maximum speed and force values are capped at 300 mm/s and 3 N in rendering also to ensure that the points remain within the triangulation convex hull. A white gaussian signal having variance equal to the interpolated signal is generated. The gaussian signal and previous vibration outputs are used to generate new vibration outputs.

VII. Numerical Evaluation

This section numerically evaluates the accuracy of the online modeling. The frequency spectrum of the rendered vibration signal is compared to that of the original vibration signal. A Hernandez-Andres Goodness-of-Fit Criterion (GFC) is used as an error metric for comparison. The frequency domain analysis compares the overall similarities in the power spectral density of the two signals. The GFC of the original signal is compared to the GFC values of the local haptic model as it is updated after each iteration.

A. Error Metric

GFC evaluates weather the reconstructed signal (rendered) actually matches the original signal. It's value ranges from zero to one, where a value of one denotes a perfect reconstruction. However, the quality of reconstruction is not linearly related to the value of GFC, i.e., a value of 0.5 would not necessarily mean half reconstruction. The human just-noticeable-difference (JND) for vibrations has been determined to be 17% for vibrations at frequencies higher than 150 Hz [38]. Therefore, a value of 0.9 or higher is considered a good GFC value as the reconstruction error remains within the human JND threshold, and users would not be able to perceive the signal mismatch [20]. GFC is calculated using the following formula:

$$GFC = \frac{\left\| \sum_i A_d\left(f_i\right) A_m\left(f_i\right) \right\|}{\sqrt{\left\| \sum_j \left[A_d\left(f_j\right)\right]^2 \right\|} \sqrt{\left\| \sum_k \left[A_m\left(f_k\right)\right]^2 \right\|}}, \quad (2)$$

where $A_d(f_i)$ and $A_m(f_i)$ are the DFT amplitudes at a frequency f_i of the measured and reconstructed signals, respectively.

In the current evaluation, the rendered signal was compared against the original signal. For evaluation purposes, the local model was built until the stopping criterion was met. Afterwards, five random AR coefficients were removed from the final model. These five models were used for evaluation of the system. The real signals associated with these five AR coefficients were stored and model building started from the beginning. The five removed AR coefficients were approximated at intervals of 2.5s during the online model update. The GFC value for comparison of the approximated signal against their real counterparts were calculated. It was hypothesized that as more data becomes available, the averaged GFC value of the five points would approach one.

A total of four textures as shown in the upper side of Fig. 4 were used to evaluate the overall framework. Data were collected from the follower side for 18 seconds from each texture surface, and a model was created for each surface. Five random AR coefficients were selected from the built model and their original interaction signals were stored. The same 18 seconds data were used to recreate the

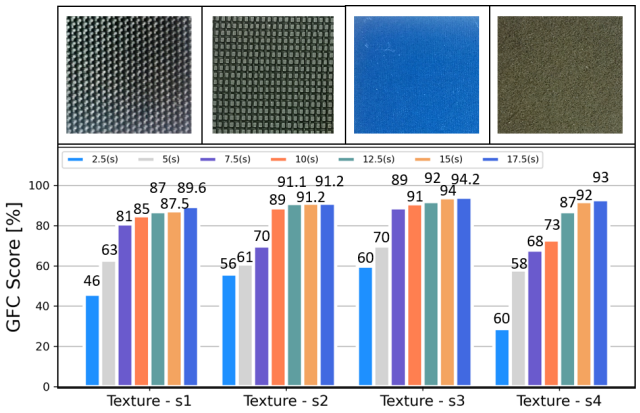

Fig. 4. Above: four texture surface samples used in the evaluation. Below: GFC values for the four different textures after every 2.5s of data collection. The values are calculated by averaging the GFC value of five AR coefficients that were excluded from the completed model.

TABLE I

TOTAL DISTANCE FOR EACH TEXTURE AFTER FIXED INTERVALS OF TIME (2.5S).

Textures	Time (s)						
	2.5	**5**	**7.5**	**10**	**12.5**	**15**	**17.5**
S1	8.74	7.31	6.11	5.38	4.67	4.56	4.41
S2	7.1	6.64	5.32	4.6	4.23	3.21	3.21
S3	8.92	7.85	6.1	4.2	3.68	3.68	3.23
S4	6.2	6.43	5.54	4.72	4.43	4	3.83

model without considering the five selected AR coefficients. The model was iteratively updated after every 2.5s of data to simulate online model update. After every update iteration (i.e., 2.5s), the signal was approximated at the selected five points. The GFC value for the five points were calculated against their original signals and averaged out.

B. Results

Figure 4 shows the result of this process. It can be seen that average GFC of the models increase as the model is updated with new data. Different textures reach the completion criterion at different times, as shown in I. The completion criterion for the models in this evaluation turned out to be 4.24 according to 1. Texture - S3 reached completion based on 10s of modeling data, S2 reached after 12.5s, S4 completed at the 15s mark, whereas, S1 did not meet the completion criterion after 18s of data. It can be seen from Fig. 4 and Table I that the GFC value reaches 90 % when the completion criterion is met. Figure 5 shows the power spectrum of the measured and synthesized signals for all four textures. The GFC values for these models are available in Fig. 4 at the 17.5s mark.

C. Discussion

The results in Fig. 4 and Table I highlight that the initial effort for establishing a haptic texture teleoperation system has shown promising outcome. However, the system works under certain limitations that are important to be addressed in future versions. First of all, the system uses streaming

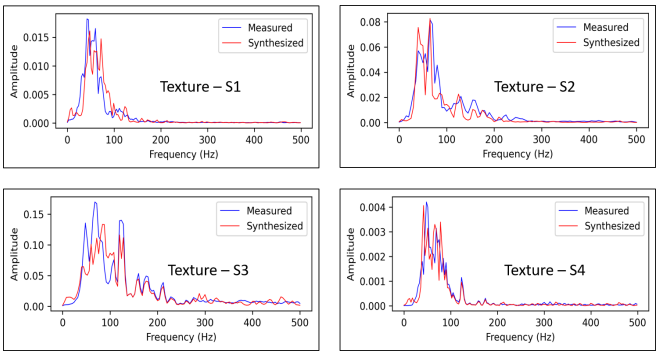

Fig. 5. Power spectrum of the original and reconstructed signals for all four completed texture models.

data until an initial model is built. In case of severe network delays, the initial interactions could be hindered and reduce the haptic experience. Second, the system switches from streaming data to synthesized data during rendering. This switching could result in a impedance mismatch and cause instability or jerky feedback. One solution would be to apply a signal smoothing or merging technique to reduce this effect. Third, the starting and completion criterion are empirically defined in the current system and as such cannot guarantee a perceptually sound model. There lies a need to carryout a comprehensive psychophysical study to find out perceptually suitable starting and stopping criterion for modeling.

VIII. CONCLUSIONS

In this study, a bilateral MMT framework for surface haptic texture sharing is introduced. From the best of our knowledge, this paper is the first attempt that applies MMT concept to haptic texture sharing. The main contribution of this work lies in the online haptic texture update framework and providing a framework for haptic texture teleoperation. All the necessary components including the hardware setup, online local texture model building, and the simulation and rendering of the texture. Rendering of texture and online update of the local model occur simultaneously at the leader side without incurring delay, loss of stability, or reduction in perception, ensuring the effectiveness of the new approach. The overall framework was tested using four different texture surfaces. The results showed that the error was reduced as the model was updated with new data.

ACKNOWLEDGMENT

This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government. [23ZS1300, Research on High Performance Computing Technology to overcome limitations of AI processing]. The authors would like to extend their gratitude towards Heather Culbertson et al. for sharing their haptic texture rendering code online. The authors are also grateful towards Arsen Abdulali et al. for providing their texture segmentation framework.

REFERENCES

[1] D. Wang, K. Ohnishi, and W. Xu, "Novel emerging sensing, actuation, and control techniques for haptic interaction and teleoperation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 624–626, 2019.

[2] N. Marturi, A. Rastegarpanah, C. Takahashi, M. Adjigble, R. Stolkin, S. Zurek, M. Kopicki, M. Talha, J. A. Kuo, and Y. Bekiroglu, "Towards advanced robotic manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy," in *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*. IEEE, 2016, pp. 1–8.

[3] V. Pruks and J.-H. Ryu, "Method for generating real-time interactive virtual fixture for shared teleoperation in unknown environments," *The International Journal of Robotics Research*, p. 02783649221102980, 2022.

[4] D.-H. Zhai and Y. Xia, "Adaptive finite-time control for nonlinear teleoperation systems with asymmetric time-varying delays," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 12, pp. 2586–2607, 2016.

[5] Z. Li, Y. Xia, D. Wang, D.-H. Zhai, C.-Y. Su, and X. Zhao, "Neural network-based control of networked trilateral teleoperation with geometrically unknown constraints," *IEEE transactions on cybernetics*, vol. 46, no. 5, pp. 1051–1064, 2015.

[6] Z. Chen, F. Huang, W. Song, and S. Zhu, "A novel wave-variable based time-delay compensated four-channel control design for multilateral teleoperation system," *IEEE Access*, vol. 6, pp. 25 506–25 516, 2018.

[7] M. Risiglione, J.-P. Sleiman, M. V. Minniti, B. Cizmeci, D. Dresscher, and M. Hutter, "Passivity-based control for haptic teleoperation of a legged manipulator in presence of time-delays," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5276–5281.

[8] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE transactions on Robotics and Automation*, vol. 5, no. 4, pp. 426–434, 1989.

[9] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, 2008.

[10] J. Song, Y. Ding, Z. Shang, and J. Liang, "Model-mediated teleoperation with improved stability," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, p. 1729881418761136, 2018.

[11] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-mediated teleoperation: Toward stable and transparent teleoperation systems," *IEEE Access*, vol. 4, pp. 425–449, 2016.

[12] I. El Rassi and J.-M. El Rassi, "A review of haptic feedback in tele-operated robotic surgery," *Journal of medical engineering & technology*, vol. 44, no. 5, pp. 247–254, 2020.

[13] D. Feth, A. Peer, and M. Buss, "Incorporating human haptic interaction models into teleoperation systems," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4257–4262.

[14] Z. F. Quek, W. R. Provancher, and A. M. Okamura, "Evaluation of skin deformation tactile feedback for teleoperated surgical tasks," *IEEE transactions on haptics*, vol. 12, no. 2, pp. 102–113, 2018.

[15] I. Sarakoglou, N. Garcia-Hernandez, N. G. Tsagarakis, and D. G. Caldwell, "A high performance tactile feedback display and its integration in teleoperation," *IEEE Transactions on Haptics*, vol. 5, no. 3, pp. 252–263, 2012.

[16] X. Xu, B. Cizmeci, A. Al-Nuaimi, and E. Steinbach, "Point cloud-based model-mediated teleoperation with dynamic and perception-based model updating," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2558–2569, 2014.

[17] S. Gallo, L. Santos-Carreras, G. Rognini, M. Hara, A. Yamamoto, and T. Higuchi, "Towards multimodal haptics for teleoperation: Design of a tactile thermal display," in *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*. IEEE, 2012, pp. 1–5.

[18] M. Aggravi, D. A. Estima, A. Krupa, S. Misra, and C. Pacchierotti, "Haptic teleoperation of flexible needles combining 3d ultrasound guidance and needle tip force feedback," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4859–4866, 2021.

[19] 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.

[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 *2012 IEEE Haptics Symposium (HAPTICS)*. IEEE, 2012, pp. 385–391.

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

[22] A. Abdulali and S. Jeon, "Data-driven rendering of anisotropic haptic textures," in *International AsiaHaptics Conference*. Springer, 2016, pp. 401–407.

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

[24] A. Abdulali, W. Hassan, and S. Jeon, "Sample selection of multi-trial data for data-driven haptic texture modeling," in *2017 IEEE World Haptics Conference (WHC)*. IEEE, 2017, pp. 66–71.

[25] S. Shin and S. Choi, "Hybrid framework for haptic texture modeling and rendering," *IEEE Access*, vol. 8, pp. 149 825–149 840, 2020.

[26] W. Fu, M. M. van Paassen, D. A. Abbink, and M. Mulder, "Framework for human haptic perception with delayed force feedback," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 2, pp. 171–182, 2018.

[27] R. Leib, A. Karniel, and I. Nisky, "The effect of force feedback delay on stiffness perception and grip force modulation during tool-mediated interaction with elastic force fields," *Journal of neurophysiology*, vol. 113, no. 9, pp. 3076–3089, 2015.

[28] S. T. Aung, Y. Ishibashi, K. T. Mya, H. Watanabe, and P. Huang, "Influences of network delay on cooperative work in networked virtual environment with haptics," in *2020 IEEE REGION 10 CONFERENCE (TENCON)*. IEEE, 2020, pp. 1266–1271.

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

[30] L. Kim, A. Kyrikou, G. S. Sukhatme, and M. Desbrun, "An implicit-based haptic rendering technique," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 3. IEEE, 2002, pp. 2943–2948.

[31] N. Zafer, "Constraint-based haptic rendering of a parametric surface," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 3, pp. 507–517, 2007.

[32] A. M. Okamura, K. J. Kuchenbecker, and M. Mahvash, "Measurement-based modeling for haptic rendering," *Haptic Rendering: Algorithms and Applications*, pp. 443–467, 2008.

[33] K. J. Kuchenbecker, J. Romano, and W. McMahan, "Haptography: Capturing and recreating the rich feel of real surfaces," in *Robotics Research*. Springer, 2011, pp. 245–260.

[34] 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, 2011.

[35] Y. Ujitoko and Y. Ban, "Vibrotactile signal generation from texture images or attributes using generative adversarial network," in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 2018, pp. 25–36.

[36] B. Willaert, J. Bohg, H. Van Brussel, and G. Niemeyer, "Towards multi-dof model mediated teleoperation: Using vision to augment feedback," in *2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012) Proceedings*. IEEE, 2012, pp. 25–31.

[37] J. S. Erkelens, *Autoregressive modelling for speech coding: estimation, interpolation and quantisation*. Citeseer, 1996.

[38] H. Pongrac, "Vibrotactile perception: examining the coding of vibrations and the just noticeable difference under various conditions," *Multimedia systems*, vol. 13, no. 4, pp. 297–307, 2008.