2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Design Octree-Based Method to Improve Model-Mediated Teleoperation in Tactile Internet

Mads Antonsen[1], Francesco Chinello[2], and Qi Zhang[1]

Abstract— **In this paper, we propose a model-mediated teleoperation (MMT) system using an octree-based model (OBM) to spatially map the environment impedance for the emerging use cases in Tactile Internet. Different from the existing just-noticeable-difference (JND) based MMT, our method avoids continuous transmission of environment impedance. Moreover, it allows the local model to generate accurate force feedback and reduces the number of model updates. Furthermore, the OBM can be deployed with or without previous knowledge of the environment. An online estimation of the OBM is proposed using a JND and a rate-of-change threshold. An offline estimation method is also proposed when the geometry and impedance parameters of the remote environment are known. In addition, a point cloud-based force rendering algorithm is tailored to use the OBM, thereby allowing the generating of force feedback for complex environments. An experiment without human-in-the-loop was conducted, showing that for an online estimated OBM, the accuracy of the force feedback was improved by up to 44 percent while using less than half the number of model updates when compared to JND-based MMT. Another experiment with a human operator interacting with a virtual environment showed that using an offline estimated OBM improves the accuracy of the force feedback and is reliable against packet loss and short temporal breakdown of the communication link.**

I. Introduction

Touch-enabled Tactile Internet is the next evolution of the Internet, enabling real-time human-to-robot interactions [1]. It aims to achieve immersive teleoperation and telerobotics over the Internet, allowing much longer physical distance between a local operator and a remote robot. A typical teleoperation system is illustrated in Figure 1. A human operator uses the local device to send position/velocity commands and to feel force feedback. The remote device executes these commands, which generate force signals from the interaction with the remote environment. Position/velocity commands and force signals are transmitted through the communication link in real-time, thereby allowing the human operator to immersively interact with the remote environment.

Long communication delay is unavoidable if the local device and remote device are geographically separated. Communication delay in the force feedback loop has been shown to negatively impact system stability and transparency [2].

This work is supported by the *eTouch: Edge Intelligence for Immersive Telerobotics in Touch-enabled Tactile Internet* project (Grant No. 1127-00339B) granted by the Danish Council for Independent Research.

[1] M. Antonsen and Q. Zhang are with DIGIT, Department of Electrical and Computer Engineering, Aarhus University, Denmark {m.m.a,qz}@ece.au.dk

[2] F. Chinello is with the Department of Business Development and Technology, Aarhus University, Denmark chinello@btech.au.dk

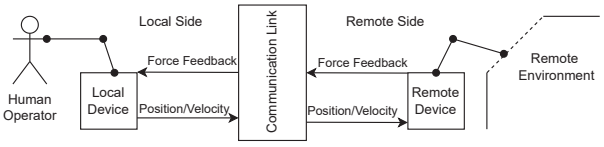

Fig. 1. Typical bilateral haptic teleoperation system adapted from [3].

Model-mediated teleoperation (MMT) is one promising approach that can address the issue of long communication delays due to large physical distance [4]. MMT uses a model at the local side to produce force feedback, thereby removing communication delay from the force feedback loop. Thus, stability and transparency can be improved. Nevertheless, it is crucial to address the challenges of MMT, including environment modeling, efficient and reliable data transmission, as well as stable control of the remote device during state transition, and stable force rendering [5]. Otherwise, they may impact system stability and transparency.

In environment modeling, model parameters are required to describe the geometry and impedance of the remote environment. Impedance parameters are estimated from the contact information (position, velocity, and force) during interactions between the remote device and the environment. Meanwhile, geometry can additionally be estimated from vision and range sensors. The environment model needs to be continuously estimated to timely detect a mismatch between the model and the remote environment, hence, fast model parameter convergence is necessary to reduce this model mismatch.

Initially, Mitra et al. proposed to estimate a simple 1-degree-of-freedom (DOF) floor model from contact information [4]. Later, the model was extended such that the location and orientation of a planar surface were estimated visual and contact information [6]. Recently, Xu et al. proposed using a point cloud to capture the complex geometry of a remote environment [7]. The point cloud was generated using depth images from a time-of-flight camera. Furthermore, environment impedance was modeled using Hooke's law and friction coefficients, estimated from contact information.

Efficient data transmission is important for MMT and it is inefficient to transmit all model parameters. Approaches for updating model parameters can be classified into time-triggered and selective transmission. Time-triggered transmission, as the name implies, triggers the transmission of model parameters based on elapsed time, whereas selective transmission only transmits selected model parameters. Se-

lective transmission can be implemented using a perceptual threshold based on the just-noticeable difference (JND) [7]. This method only triggers the transmission of model parameters after the change is large enough to be perceivable by the human operator. Moreover, a method combining selective and time-triggered transmission has also been proposed [8].

The current work on MMT often assumes an environment where changes are spatially infrequent [7]. However, this assumption may fail if an application requires the operator to repeatedly move the remote device between materials with different physical properties. This leads to an increasing number of model mismatches which triggers model updates. Ideally, the spatial information of estimated impedance should be considered. Furthermore, current work cannot easily include existing information of the environment, which depending on the applications, may be partially or completely known before teleoperation starts. Such applications could be inspection and maintenance of known, but inaccessible, and dangerous environments. Including this information could greatly improve model accuracy and thereby system transparency.

In this paper, we made the following contributions: (I) We propose an impedance mapping method based on octrees for continuously mapping the impedance parameters of a static environment, namely an octree-based model (OBM). Different from previous MMT which only uses the most recent estimated impedance parameters, the OBM allows for mapping the impedance parameters while exploring the remote environment, thereby creating a more comprehensive model of the remote environment. This helps to reduce the mismatch between the local model and the remote environment during the transition between materials. Furthermore, unlike previous MMT methods, this method also allows for prior knowledge to be integrated into the model, thereby improving initial model accuracy. (II) We extend the perceptual threshold based on JND, by adding a rate-of-change threshold to ensure stability of the impedance parameters, thereby minimizing the chance of mapping mismatched impedance parameters. Lastly, the OBM is combined with a point cloud-based force rendering algorithm to generate force feedback for the local device.

The rest of this paper is structured as follows. Section II introduces the octree-based model for model-mediated teleoperation. Section III describes two experiments and the results. Section IV concludes this paper and outlines future work.

II. Octree-Based Model for Model-Mediated Teleoperation

Figure 2 illustrates the proposed system. The modeling block utilizes the measured force and position signals from the remote device to estimate the impedance parameters of the remote environment. Subsection II-B provides a detailed description of the octree-based model (OBM) in which impedance parameters are mapped.

On the remote side, the updating controller determines when it is necessary to map the impedance parameters to

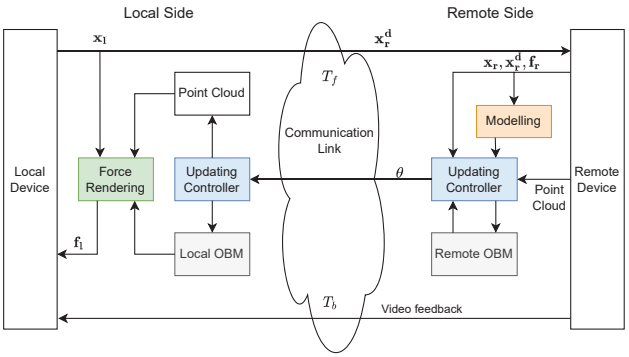

Fig. 2. Overview of proposed MMT system, where T_f is forward delay; T_b, backward delay; $\mathbf{x_l}$, local position; $\mathbf{f_l}$, local force; $\mathbf{x_r}$, remote position; $\mathbf{x_r^d}$, desired remote position; $\mathbf{f_r}$, remote force; θ, impedance and geometry model parameters.

the remote OBM. Any update to the remote OBM is subsequently transmitted to the local side, where the local OBM is also updated. Simultaneously, the remote-side updating controller ensures that the geometric model is kept updated. In this paper, we utilize a point cloud to represent the environment geometry, where each point corresponds to a location on the surface of the remote environment.

A. Modeling

For this paper, we use a linear spring model to model the impedance of the remote environment, as it can be quickly estimated. Force and position signals are measured while the end-effector of the remote device interacts with the remote environment. The difference between its actual position and the desired position is used as the penetration depth. Using these signals and Hooke's law, the stiffness of a linear spring can be estimated

$$\Delta d = ||\mathbf{x_r} - \mathbf{x_r^d}||, \tag{1}$$

$$k_n = \frac{||\mathbf{f_r}||}{\Delta d}, \tag{2}$$

where k_n is the stiffness of a linear spring and $||\mathbf{f_r}||$ is the force magnitude of the remote device. Δd is the penetration depth estimated from the difference between the desired position $\mathbf{x_r^d}$ and actual position $\mathbf{x_r}$ of the remote device.

The stiffness k_n is estimated from the last 100 measured data samples using least squares similar to [7]. Estimation is only active while the remote device is in contact with the environment. The force $\mathbf{f_r}$ must exceed the contact threshold f_c, to ensure that the stiffness is estimated only if the remote device is in contact with the remote environment.

B. Octree-Based Model

An octree is a tree structure that can be used for spatial partitioning of space, such as 3D mapping in robotic applications [9]. The first node in an octree is called a root node, while the nodes at the end of the octree are the leaf nodes. The spatial volume covered by a node is the *octant*, which consists of a center point and an extent. Subdividing a leaf node, eight new child nodes are created, with the corresponding octants having half the extent of the parent,

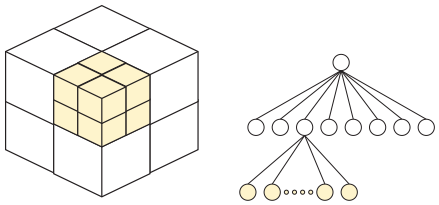

Fig. 3. (Left) spatial volumes described by the octants of the octree. (Right) the corresponding nodes of the octree.

thereby allowing the eight new octants to fit within the parent octant as illustrated in Figure 3.

The octree-based model uses an octree for spatial partitioning and mapping regions of the environment with corresponding impedance parameters. Large regions of the environment with similar impedance parameters can be described using a few but large octants, and when necessary, the octree can be subdivided to accurately map small features, such as the border between materials. This allows for efficient mapping as the resolution can be varied to fit the environment.

The OBM can be estimated either online or offline. Online estimation does not require any prior knowledge of the remote environment. Offline estimation requires prior knowledge of the remote environment; that is, model parameters for the geometry and impedance are already known. Offline estimation can provide more accurate force feedback, but the required prior knowledge may not always be available.

Online Estimation: In MMT, impedance parameters are commonly updated based on a perceptual threshold using JND. Parameter updates are performed when the difference between the last transmitted and the newly estimated parameters exceeds the JND threshold. However, exceeding this threshold does not guarantee convergence, that is, risking mapping a mismatched model in the OBM. To mitigate this phenomenon, a rate-of-change (ROC) threshold is applied. Equation (3) summarizes the new updating rule.

$$\text{Update} = \begin{cases} \text{Yes}, & \text{if } \left|\frac{k_O - k_n}{k_O}\right| > \Delta_k \text{ and } \delta < \delta_k \\ \text{No}, & \text{else}, \end{cases} \tag{3}$$

where k_n is the estimated stiffness from (1), k_O is the mapped stiffness at remote position $\mathbf{x_r}$, Δ_k is JND threshold for stiffness, δ_k is the ROC threshold, and δ is the current ROC. Previous work found JND for stiffness to be 23% [10]. Current ROC δ is based on the last 5 stiffness estimates,

$$\delta = \frac{1}{5}\sum_{i=1}^{5}\left|\frac{k_{n-i+1}}{k_{n-i}} - 1\right| 100. \tag{4}$$

When the OBM should be updated with new impedance parameters, the leaf node must either be updated or subdivided depending on its current depth and parameters. When a leaf node is subdivided, all the child nodes inherit the new parameters. Each update is transmitted to the local side, where they are applied to the local OBM.

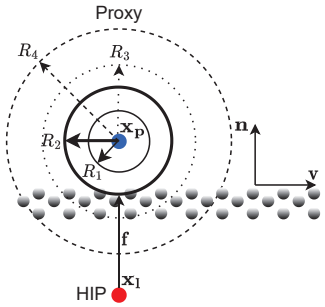

Fig. 4. Proxy (blue) with four radii outside the point cloud and HIP (red) penetrating into the point cloud. Force $\mathbf{f}$ is calculated from stiffness k_O and penetration depth of HIP.

Offline Estimation: The OBM can be estimated offline with prior information about the remote environment. Given a set P containing polygons describing known surfaces and a set I describing impedance parameters of known materials, where each polygon in P corresponds to impedance parameters in I, then these impedance parameters can be integrated into the OBM.

The process involves subdividing the OBM to the desired resolution, determining octants that intersect with each polygon in P using the separate axis theorem [11], and then assigning the corresponding impedance parameters from I to the octants.

C. Point Cloud-based Force Rendering

The octree-based model (OBM) is combined with a geometric model for force rendering, as the OBM only describes the environment impedance. As previously mentioned, the geometric model for our method is based on a point cloud, and the force rendering is performed similarly to [12].

Force rendering with a point cloud is based on the fast plane detection method proposed in [13]. This method attempts to estimate a local plane at the point of interaction. The method uses a haptic interaction point (HIP) and proxy for collision detection and force rendering. The method is modified to use four radii, R_1, R_2, R_3, and R_4 around the proxy, to determine collision with the point cloud. The four radii, and an interaction between the HIP, proxy, and a point cloud are illustrated in Figure 4.

Proxy movement: The HIP represents the current position of the local device $\mathbf{x_l}$ and is allowed to pass through the point cloud. The proxy $\mathbf{x_p}$ is restricted to only move along the point cloud. Movement of proxy is performed in small steps and as many times as possible within 1 ms (1 kHz force feedback loop frequency). Each proxy step will be performed according to one of the following rules:

1) The proxy is free if there are no points inside R_2. Proxy moves one step towards the HIP, where the step is defined by

$$\mathbf{s} = s_0 \cdot \mathbf{u}, \tag{5}$$

where $\mathbf{s}$ is the step, s_0 is the step size, $\mathbf{u}$ is the direction

from proxy $\mathbf{x_p}$ to HIP $\mathbf{x_l}$ given by

$$\mathbf{u} = \frac{\mathbf{x_l} - \mathbf{x_p}}{||\mathbf{x_l} - \mathbf{x_p}||}. \tag{6}$$

2) The proxy is entrenched if any points are within R_1. The proxy must move one step in the direction along the surface normal to break entrenchment. The step $\mathbf{s}$ is given by

$$\mathbf{s} = s_0 \cdot \mathbf{n}, \tag{7}$$

where $\mathbf{n}$ is the surface normal estimated from the points that are within R_3. The method for estimating the surface normal is given as

$$\mathbf{n} = \frac{1}{K} \sum_{i=1}^{K} \frac{\mathbf{x_p} - \mathbf{x_i}}{||\mathbf{x_p} - \mathbf{x_i}||}, \tag{8}$$

where K is the number of points inside R_3, $\mathbf{x_p}$ is the proxy position, and $\mathbf{x_i}$ is the position of ith point inside R_3.

3) The proxy is in contact and points are inside R_2, but HIP is outside the point cloud. HIP is outside point cloud if the angle between direction $\mathbf{u}$ and surface normal $\mathbf{n}$ is larger than 90 degrees. The angle is given by

$$\phi = \mathbf{u} \cdot \mathbf{n}, \tag{9}$$

where ϕ is the angle. Proxy is moved a step towards the HIP based on (5).

4) The proxy is in contact if there are points inside R_2, and is allowed to move in any direction along the estimated surface, i.e. orthogonal to the surface normal. The direction along the surface is found by the vector rejection of $\mathbf{u}$ from $\mathbf{n}$

$$\mathbf{v} = \mathbf{u} - \frac{\mathbf{u} \cdot \mathbf{n}}{\mathbf{n} \cdot \mathbf{n}} \mathbf{n}, \tag{10}$$

where $\mathbf{v}$ is the direction of the proxy to step along the surface. The step is calculated similarly to previous cases $\mathbf{s} = s_0 \cdot \mathbf{v}$.

For each proxy step in the haptic frame, intersection testing between each point and the radii must be performed, which can be costly for a large number of points. Therefore, all points are tested against the radii R_4 at the beginning of a haptic frame, and the points within R_4 are used for the proxy movement steps. A radii $R_4 = 4R_3$ is used.

Force Rendering: Force feedback must be rendered if the proxy is in contact with the point cloud. Contact occurs if points are within R_2 when the haptic frame ends. The position difference between contact point $\mathbf{x_c}$ and the HIP $\mathbf{x_l}$ is used as the penetration depth. The contact point is assumed to be the point where the direction vector between HIP and proxy makes contact with R_2. The contact point is estimated as

$$\mathbf{x_c} = \mathbf{x_p} + \frac{R_2}{||\mathbf{x_l} - \mathbf{x_p}||}(\mathbf{x_l} - \mathbf{x_p}). \tag{11}$$

Force feedback in (12) is calculated based on Hooke's law

$$\mathbf{f_l} = \mathbf{u} \cdot k_O \cdot (||\mathbf{x_l} - \mathbf{x_p}|| - R_2), \tag{12}$$

where $\mathbf{f_l}$ is force feedback rendered by the local device, $\mathbf{u}$ determines the direction of the force, k_O is stiffness from local OBM at contact point $\mathbf{x_c}$, and the last part describes the penetration depth, which assumes contact at R_2.

III. Experiments and Results

Two experiments are conducted to test model accuracy, the number of updates, and the robustness to packet loss of the proposed system. The first experiment uses a simulation without human-in-the-loop (HITL) to test the convergence and the number of necessary model updates. The second experiment has a human operator executing a line-following task using a teleoperation system with a virtual remote side.

A. Experiment without Human-in-the-Loop

In this experiment, we explore the number of model updates required to reach a certain model accuracy for the perceptual threshold based on the just-noticeable difference (JND), and the proposed octree-based model (OBM). The OBM is tested with and without the rate of change (ROC) threshold, to analyze the impact that the threshold has.

A virtual environment (VE) and a virtual device are simulated with Chai3D which is a haptics simulation framework [14]. The VE consists of a flat plane, which is subdivided into regions with varying stiffness as illustrated in the left side of Figure 5. The plane is assumed to only consist of a stiffness and no communication delay is employed. To generate the OBM an ROC threshold $\delta_k = 0.002$ is used with the JND threshold being 23%.

A set of 1200 interactions is created by uniformly distributing the interaction's starting point and ending point on the VE plane, resulting in a linear trajectory. The virtual device executes an interaction by moving between the two points with a constant velocity $v = 0.75m/s$ and each interaction is then performed in sequence. Furthermore, the set of interactions is permutated 1000 times to change the sequence of performed interactions, and each permutation is executed to average the results.

In [15], the perceptual mean-squared error (PMSE) was proposed to compare two haptic signals and is shown in (13). We use it to compare the rendered force with the measured force to evaluate model accuracy.

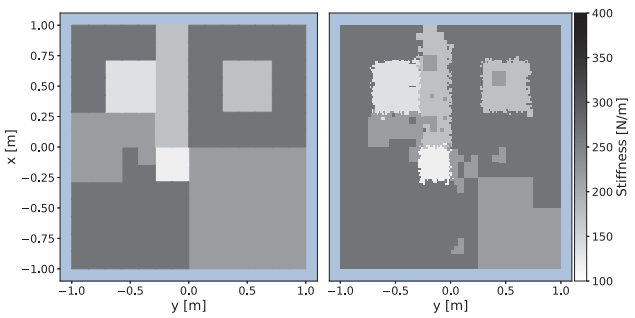

Fig. 5. (Left) plane used for the VE. (Right) Top-down illustration of the OBM generated from 1200 interactions at $v = 0.75m/s$. Stiffness is visualized as a greyscale color.

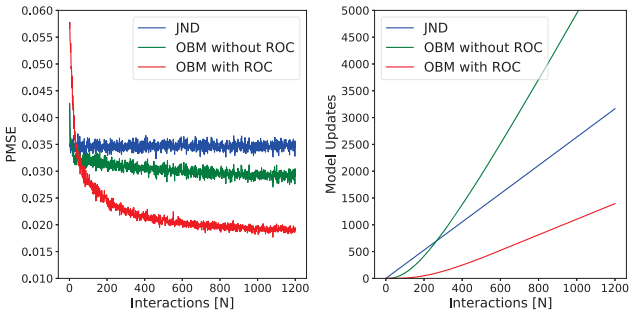

Fig. 6. (a) PMSE comparison and (b) the number of model updates performed by the updating controller with end-effector's velocity v.

$$\text{PMSE} = \frac{c^2}{N}\sum_{i=1}^{N}\left(\ln\left|\frac{f_r^z(i)}{f_l^z(i)}\right|\right)^2, \tag{13}$$

where N is the number of samples in a trajectory, f_l^z and f_r^z are the z-axis force signals measured from the local device and remote device respectively, and c is a scaling constant. We use the same scaling constant $c = 1$ similarly to [8]. The lower the PMSE value is, the higher the model accuracy is.

Figure 6 (a) shows the PMSE and Figure 6 (b) the number of model updates for JND (blue line), OBM without rate of change (ROC)(green line), and OBM with ROC (red line). Looking at the results in Figure 6 we see that JND has a constant PMSE over all the interactions, as the method does not map the environment as it is being explored. Meanwhile, the number of model updates is linearly increasing as each perceptual change in impedance has to be transmitted. Looking at OBM without ROC we see an improvement of $\sim 14\%$ in PMSE, but at the cost of a significantly higher number of model updates. OBM with ROC reaches a PMSE that is $\sim 42\%$ and $\sim 33\%$ lower than without ROC and JND, respectively. This is especially impressive when considering its lower number of model updates. From this, we can see that the OBM should use an ROC threshold to correctly and efficiently map the environment.

This experiment showed that OBM with ROC could reach a higher accuracy than JND when the OBM in online estimation. Meanwhile, fewer model updates are needed to reach better accuracy with ROC. However, as indicated by the rendered OBM in Figure 5 on the right side, we see the method for online estimation can be improved as it does not completely map the border between regions. A partial explanation for this is the virtual environment itself. Some of the materials only cause the perceptual threshold to trigger in one direction i.e., low stiffness to high stiffness but not in reverse. One solution to solve this issue may be to use a lower perceptual threshold.

B. Experiment with Human Operator

In this experiment, we analyze and compare the robustness and model accuracy of an offline estimated OBM with respect to packet loss and model mismatch, to JND with time-triggered updating and without time-triggered updating. JND

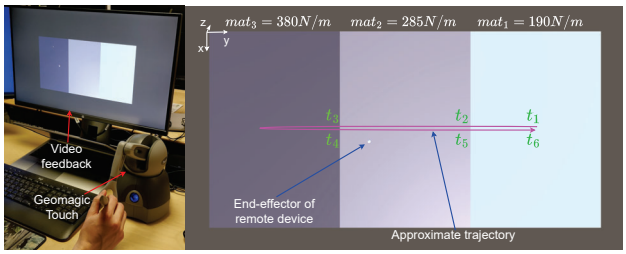

Fig. 7. (Left) local device for force feedback and display for video feedback. (Right) virtual remote side with approximate trajectory, virtual remote device, and VE showing stiffness for material mat_1, mat_2, and mat_3.

without time-triggered updating only updates impedance upon exceeding the perceptual threshold, thereby achieving a very low packet rate but is also more susceptible to lost packets. JND with time-triggered updating trades higher packet rates for improved model accuracy and robustness to lost packets.

This experiment uses a virtual remote side implemented using Unity for visual and point cloud data, and Chai3D for physical interactions between the virtual remote device and virtual remote environment. To emulate the noise model of a physical force sensor, we add artificial sensor noise following a Gaussian distribution is added to the force sensor of the virtual remote device. Meanwhile, a Geomagic Touch haptic device is used as the local device providing a means for the operator to feel the force feedback. The virtual environment is shown in Figure 7 and illustrates the executed trajectory and its six transitions $(t_1, \ldots, t_6)$, where a transition is either a change between materials or making/breaking contact with the environment surface.

For each experimental case, constant forward and backward delays of $T_f = T_b = 100$ ms are applied. For the experimental cases with OBM, a 5 cm misalignment on the y-axis and a 5% error in mapped environment stiffness is introduced to cause model a mismatch and trigger model updates. For experimental cases of JND, a 23% threshold for stiffness updating is used. Based on previous work by Xu et al. we use a 150 ms time updating threshold for the experimental cases with time-triggered updated [8]. For the non-ideal experimental cases, artificial unreliable communication conditions are introduced at transition t_3, as a 500 ms breakdown of the communication link from the remote side to the local side.

Results for ideal conditions are shown in Figure 8 (a)-(c). We see that accurate local force can be rendered using the OBM, even though a model error and misalignment between the OBM and remote environment is introduced. JND is less accurate than OBM as it is not able to capture the entire impedance change seen at transition t_2 and t_4. With time-triggered updating JND becomes more accurate as it will eventually capture this impedance change, however, this is at the cost of more model updates increasing the load on the communication link.

Results for non-ideal conditions are shown in Figure 8 (d)-(f). Even though model updates are lost at transition t_3,

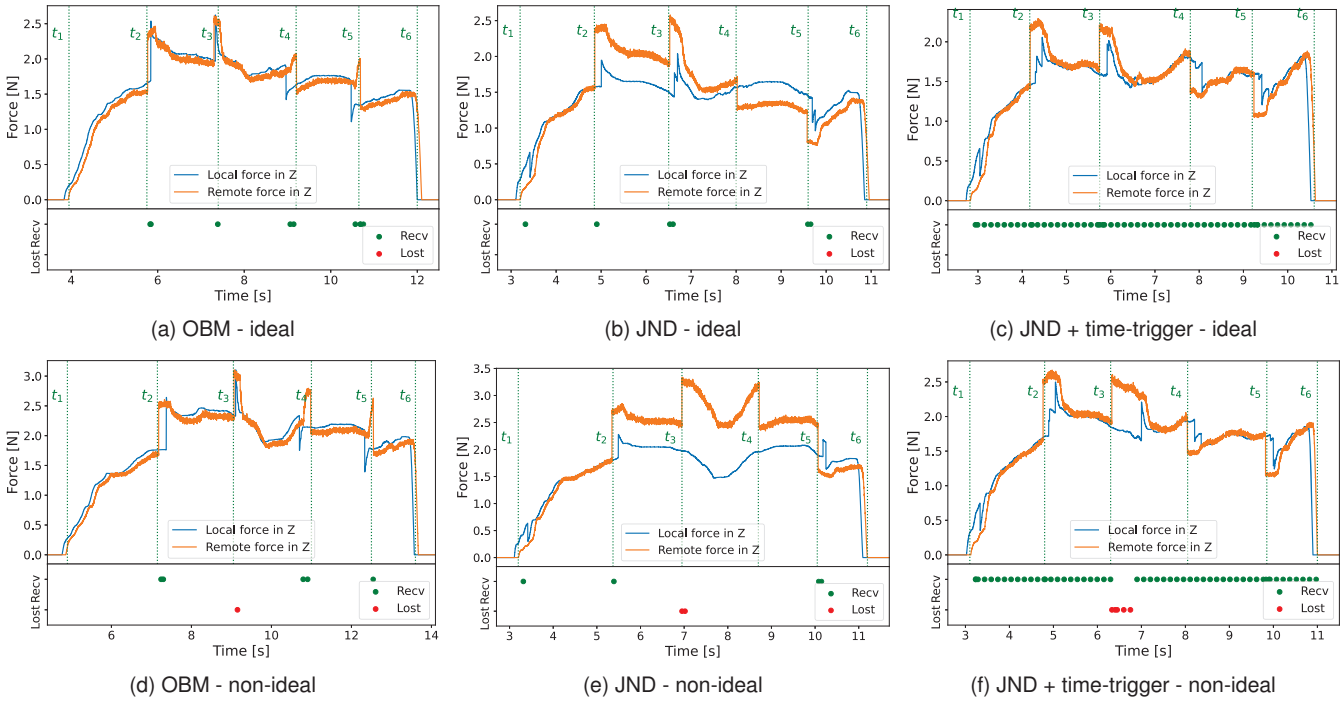

Fig. 8. Force on Z-axis for six experimental cases. At t_1, the remote device makes contact with material mat_1 in the environment. At t_2, the remote device crosses from the soft mat_1 to the hard mat_2. At t_3, the remote device crosses from the hard mat_2 to the harder mat_3. At t_4, the remote device crosses from the harder mat_3 to the hard mat_2. At t_5, the remote device crosses from the hard mat_2 to the soft mat_1. At t_6, the remote device ends contact with the environment. Lost (red) and received (green) packets are shown at the bottom section of each plot for each experimental case.

OBM still performs well under non-ideal conditions as also seen in Table I. The main reason for this is that the local model maps the impedance parameters' spatial relationship in the remote environment. On the other hand, JND does not use a sophisticated model, hence, must propagate any perceivable change of the estimated impedance as a model update. This makes it more susceptible to packet loss as seen in Figure 8 (e) after transition t_3. JND with time-triggered updating alleviates this problem by continuously transmitting the estimated impedance, hence, the local side can timely catch the model updates.

The results are summarized in Table I using perceptual mean squared error introduced in (13) and the absolute maximum error.

TABLE I

PERCEPTUAL MEAN-SQUARE ERROR AND ABSOLUTE MAX ERROR, ADJUSTED FOR COMMUNICATION DELAY.

	OBM	JND	JND + time-trigger
Perceptual Mean-squared Error			
Ideal	0.0081	0.0576	0.0316
Non-Ideal	0.0096	0.0934	0.0379
Absolute Max Error [N]			
Ideal	0.7791	1.0471	0.8153
Non-Ideal	1.3489	1.0525	0.9554

The PMSE in Table I shows that OBM in comparison to JND with time-triggered updating, achieves up to 3.9 times improvement in accuracy of force feedback under both the ideal and non-ideal conditions. Although, as the human operator performing each experimental case, the task length and number of samples differ between experimental cases, so the PMSE reported in Table I should be taken with some reservation. As for the reported absolute max error, it depends on how much force the operator is exerting and the model mismatch during a transition. Nevertheless, it shows a scheme for safe and stable control of the remote device is necessary to avoid dangerous interactions during a model mismatch.

IV. CONCLUSION

In this paper, we propose an octree-based model (OBM) for spatially mapping the physical properties of a remote environment for model-mediated teleoperation (MMT) in Tactile Internet. We propose a method for online estimation and offline estimation of the OBM. A method for force rendering using point clouds is tailored to use the OBM, thereby allowing force feedback in complex environments. The experimental results show improved accuracy and low packet rate using the proposed OBM in comparison with the existing JND-based methods. OBM also demonstrates its robustness to packet losses.

Future studies will evaluate system performance using a physical remote side and more experimental conditions such as non-static environments, stochastic network conditions etc. should also be introduced to conduct a more comprehensive evaluation. Furthermore, a subjective evaluation of the proposed system is required. Lastly, the use of visual information from camera sensors can be explored to improve convergence for the online estimated OBM and to handle non-static environments.

REFERENCES

[1] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5g-enabled tactile internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, 2016.

[2] D. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, 1993.

[3] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *IEEE Spectrum*, vol. 4, no. 10, pp. 81–88, 1967.

[4] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, 2008. [Online]. Available: https://doi.org/10.1177/0278364907084590

[5] 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.

[6] 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*, 2012, pp. 25–31.

[7] 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.

[8] X. Xu, S. Chen, and E. Steinbach, "Model-mediated teleoperation for movable objects: dynamics modeling and packet rate reduction," in *2015 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, 2015, pp. 1–6.

[9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr 2013. [Online]. Available: https://doi.org/10.1007/s10514-012-9321-0

[10] L. A. Jones and I. W. Hunter, "A perceptual analysis of stiffness," *Experimental Brain Research*, vol. 79, pp. 150–156, 1990.

[11] T. Akenine-Möller, "Fast 3d triangle-box overlap testing," in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 8–es. [Online]. Available: https://doi.org/10.1145/1198555.1198747

[12] X. Xu, B. Cizmeci, and E. Steinbach, "Point-cloud-based model-mediated teleoperation," in *2013 IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE)*, 2013, pp. 69–74.

[13] F. Rydén, S. Nia Kosari, and H. J. Chizeck, "Proxy method for fast haptic rendering from time varying point clouds," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2614–2619.

[14] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, J. Warren, O. Khatib, and K. Salisbury, "The chai libraries," in *Proceedings of Eurohaptics 2003*, Dublin, Ireland, 2003, pp. 496–500.

[15] R. Chaudhari, E. Steinbach, and S. Hirche, "Towards an objective quality evaluation framework for haptic data reduction," in *2011 IEEE World Haptics Conference*, 2011, pp. 539–544.