



Deep encoder–decoder network based data-driven method for impact feedback rendering on head during earthquake

Joolekha Bibi Joolee¹ · Mohammad Shadman Hashem¹ · Waseem Hassan¹ · Seokhee Jeon¹

Received: 14 February 2023 / Accepted: 10 October 2023 / Published online: 18 January 2024
© The Author(s) 2024

Abstract

In safety training simulators, realistic haptic feedback is essential to make people construct accurate situation awareness through experiencing. In this regard, this paper presents a new and innovative system that provides the haptic experience of falling objects on user's head during an earthquake. Special focus was on the accurate reproduction of impact feedback when various objects fall on the head. To this end, we propose a novel data-driven approach. This approach first collects 3-axis acceleration signals during real collision under several impact velocities. Afterward, 3D acceleration data is abstracted to a 1D acceleration profile using our novel max–min extraction approach. The impact signal for an arbitrary velocity is interpolated using a deep convolutional bidirectional long short-term memory encoder–decoder model. Rendering hardware is also implemented using high performance voice-coil vibrotactile actuator. Numerical and subjective evaluations are carried out to evaluate the performance of the proposed approach.

Keywords Impact feedback · Data-driven approach · Convolutional bidirectional long short-term memory encoder–decoder · Max–min extraction

1 Introduction

An earthquake is a devastating natural disaster that occurs suddenly and with great intensity. On average, each year, 1676 earthquakes with a magnitude of five or higher occur worldwide (Development of the number of earthquakes 2020). Earthquakes result in destructive losses, affecting both property and personnel. The extensive damage and injuries underscore the importance of earthquake preparedness and protection. Moreover, current earthquake forecasting methods are not advanced enough to provide precise advance warnings. Consequently, people have limited

opportunities to evacuate before an earthquake occurs. Until a time when earthquakes can be predicted well in advance, one of the possible solutions for reducing personnel injuries is to introduce effective earthquake simulation exercises (Li et al. 2017). In these exercises, people can experience simulated earthquakes and learn to respond to different kinds of situations that may occur.

Traditionally in schools, earthquake safety has been taught through exercises and emergency safety drills. This approach faces the issue of lacking standardization (Ramirez et al. 2009). Since natural disasters are relatively rare occurrences, many people do not take safety exercises seriously. Reports indicate that many of these practices fail to account for various factors that may arise in real emergency situations. Consequently, the exercises conducted in schools may not be sufficiently effective for earthquake safety training (Li et al. 2017; Ramirez et al. 2009). Several scenarios can occur in an earthquake event. One of the most common scenarios, even in lighter earthquakes, is the falling of objects from tabletops, cupboards, or ceilings. Sometimes in the spur of the moment, people do not realize the extent of this danger and may put themselves in harm's way. People easily get bewildered and cannot comprehend the danger of falling objects. However, training people through simulated exercises to

✉ Seokhee Jeon
jeon@khu.ac.kr
Joolekha Bibi Joolee
julekhajulie@gmail.com

Mohammad Shadman Hashem
ayon7019@gmail.com

Waseem Hassan
waseem.h@khu.ac.kr

¹ Department of Computer Science and Engineering, Kyung Hee University, 446-701, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, Republic of Korea

respond properly to such situations can significantly reduce accidents caused by falling objects.

Various researchers have attempted to enhance the realism of earthquake training simulations and better replicate emergency situations. For this purpose, different virtual reality (VR) systems were designed where a user can control an avatar in a virtual environment during earthquakes, e.g., Li et al. (2017), Gong et al. (2015), Liang et al. (2018), Lovreglio et al. (2018), Xu et al. (2019). Different kinds of exercises and drills were implemented and users were asked to practice the scenarios in the virtual environment. Nevertheless, all these efforts did not provide any haptic feedback to the user, instead, they relied on the visual and aural senses only. The inclusion of haptics, in addition to visual and auditory elements, could greatly enhance realism. Consequently, incorporating haptic feedback can enhance the interaction with the system and improve the overall experience during virtual earthquake safety drills. In earthquake readiness training, trainees need to experience and respond to various physical sensations and impacts. Realistic haptic feedback can induce stress and emotional responses similar to those experienced during real earthquakes. This aspect is crucial for training individuals to manage their emotions, stay focused, and make critical decisions under stress, as earthquake situations often involve high-pressure scenarios.

In this regard, to provide the experience of real earthquake scenarios that can be used during safety-training simulated exercises, we propose a new and innovative system that renders realistic impact feedback of falling objects on head during an earthquake in the VR environment. The system is constructed to provide rich impact feedback to make users construct a real-like experience that may help users to cope with the situation while keeping them safe. This is one of the important merits of a VR-based training system where a trainee is provided with an in-advance experience of a dangerous situation without actually putting him/her in real danger. It is known that feedback on head is well suited for safety training exercises and guidance since the head is sensitive to mechanical stimuli (Gilliland and Schlegel 1994; Kaul and Rohs 2017). Feedback on the head also greatly enhances realism (Kaul and Rohs 2017). In this work, impact feedback is provided in the form of a 1D impact acceleration profile, which conveys a clear sensation of collision with a strong, short, and clean impulse of force. Realistic impact feedback includes a small vibration signal along with the impulse signal (Park et al. 2019; Lopes et al. 2015).

In general, haptic feedback is modeled and rendered based on two approaches: physics-based parametric approach [e.g., Park et al. (2019), Park and Choi (2017)] and data-driven approach [e.g., Culbertson et al. (2014), Yim et al. (2016), Osgouei et al. (2020)]. Both approaches have their own benefits and limitations. In the physics-based approach, haptic responses are defined by the coefficients of physics-based

parametric models and simulated using the parametric model during rendering. For instance, vibrotactile feedback is produced by applying an exponentially decaying sinusoidal model for collision events (Park and Choi 2017). While this approach is flexible and fast, its simulation accuracy is limited by the applied physics model. The model is usually simplified for efficiency, which often results in reduced realism of the feedback. In contrast, a data-driven approach generates feedback based purely on recorded signals without considering underlying physical principles. It has the virtue of high realism in feedback, but lower flexibility is one of its drawbacks. The approach has been successful in different haptic simulations, e.g., Culbertson et al. (2014), Yim et al. (2016), Osgouei et al. (2020), Abdulali and Jeon (2016), Abdulali et al. (2018), Shin and Choi (2020). One major limitation with these methodologies is their inability to fully incorporate phase information. To address this issue, deep spatio-temporal network (DSTN) (Joolekha and Jeon 2022) is proposed for haptic texture modeling and rendering.

In this work, we propose a novel deep network based data-driven approach in order to render the falling object impact feedback. This approach first collects time-series acceleration profiles produced when an object collides with a rigid and spherical object. For various objects, data-collection is done for different falling heights, i.e., different impact velocities. Afterward, we introduce a max–min extraction approach that converts the captured 3D acceleration signals to a 1D impact acceleration profile. Additionally, experiments are performed to demonstrate the effectiveness of the proposed max–min extraction approach over the state-of-the-art DFT321 approach (Park and Kuchenbecker 2019). The core part of the proposed data-driven approach is the deep learning-based signal interpolation algorithm. To provide realistic impact feedback, the approach should accurately estimate the acceleration profile for any arbitrary impact velocity. Because the acceleration profile changes with variations in velocity, and it can result in a different feedback. Therefore, to estimate the acceleration profile accurately, we formulate a multivariate sequential time series prediction model, which predicts the intermediate time-series acceleration profile based on the collected two adjacent acceleration profiles and given object velocity. Recent progress in deep learning, particularly Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models present some useful insights to solve sequence predicting problems. Motivated by these works, we design a deep Convolutional Bidirectional Long Short-Term Memory (ConvBi-LSTM) encoder–decoder framework to interpolate the acceleration signal appropriately for the given object velocity. For the given object velocity, the proposed model takes two neighboring impact acceleration profiles from the database (i.e., one with immediate lower impact

velocity and one with immediate higher impact velocity) as input along with the given velocity and predicts a new impact acceleration profile. The relationship among these signals is trained using the collected ground truth data. To the best of our knowledge, this is the first attempt to render realistic impact feedback using a deep learning-based data-driven approach. We adopt this deep learning-based interpolation over simple linear interpolation to handle signal complexity. In particular, our method successfully conducts interpolation between profiles with sparse input, full time series signal, and high-frequency components, which is tricky with simple linear interpolation in the time domain.

Furthermore, we also design a physics-based approach as a baseline method to compare the performance with the data-driven approach. In the physics-based approach, impact is represented as the energy of a collision between a person's head and the falling object. The energy computation takes velocity as well as mass of the falling object as input, which is then used to estimate the amplitude of the impact. The two approaches are implemented on multiple rendering actuator setups, i.e., voice coil actuators and push–pull solenoid. The proposed data-driven approach is evaluated numerically as well as perceptually with human participants. The major contributions of this work are summarized as follows.

- We designed customized interfaces for impact feedback rendering on head. Three different actuators, i.e., vibro-transducer, haptuator, and push–pull solenoid are used.
- A virtual environment combined with the haptic interface that can be utilized for earthquake experience and safety training.
- To convert the three-axis acceleration signals into 1D acceleration profile efficiently and effectively, we present a data conversion approach called max–min extraction.
- Deep learning based data-driven approach for impact feedback rendering is presented, which employs the deep ConvBi-LSTM encoder–decoder framework for interpolating the acceleration profile.
- Numerical evaluation with four different real objects to show the effectiveness of the proposed ConvBi-LSTM encoder–decoder-based interpolation. In addition, user studies are conducted for demonstrating the subjective and perceptual performance of the data-driven and baseline physics-based approaches.

The rest of the paper are organized as follows. Prior studies performed in the field of haptic feedback on the head, haptic feedback for safety purposes, and impact feedback-based approaches are discussed in Sect. 2. In Sect. 3, system and rendering hardware are presented. Data-driven impact feedback is explained along with the details of proposed deep encoder–decoder model in Sect. 4. Experimental evaluations

are carried out to estimate the performance of both data-driven and baseline physics-based approaches in Sect. 5. Finally, we summarize our contribution in Sect. 6.

2 Related works

This section discusses the techniques and methodologies that are used for haptic feedback on the head, haptic feedback for safety purposes, and feedback based on the impact signal.

2.1 Haptic feedback on head

Haptic feedback on the head is mainly provided for games, navigation, and guidance systems. A majority of these applications considered haptic feedback as a means of another information transfer channel, so they lack realism. For instance, Kaul and Rohs (2017) proposed a system for intuitive haptic guidance on the head through moving tactile cues (i.e., virtual objects, shockwaves, and particles) in both VR and AR environments. In their work, coin-type vibration motors were utilized in three concentric ellipses for feedback rendering. Similarly, Gallo et al. (2020) developed a navigation system based on head scanning and provided simple binary feedback. In their approach, the vibration signal is provided as a penalty when the runner looks in the wrong direction at an intersection. Haptic feedback on the head also can be rendered using various modalities, i.e., vibrotactile feedback, pressure-based feedback, force-based feedback, or a combination of them. In Gunther et al. (2020), pneumatic pressure feedback is provided to the user's head in a Volleyball game in VR, where 250 kPa pressure was applied through an air compressor and solenoid valves are used to control the actuators. In contrast, Tsai et al. rendered 2.5D impact feedback on the head during a boxing game and a goalkeeping game in the VR environment (Tsai et al. 2019). In their work, they designed impact actuators using a DC motor, a servo motor, and a mechanical brake. The actuators were placed in three different areas of the head to provide the normal directional impact and tangential impact respectively. However, in their work, they rendered fixed force-based impact feedback, whereas our work provides more sophisticated dynamic impact feedback.

2.2 Haptic feedback for safety drills

Various haptic guidance systems are introduced for navigation, gaming, training, etc. In this paper, we are interested in the training/guidance systems used for safety, such as interactive VR fire extinguisher (Seo et al. 2019), and assisted driving (Girbés et al. 2016; Seo et al. 2019) designed a fire extinguisher with haptic feedback in the VR environment. In their work, kinesthetic feedback is provided using

a pneumatic actuator to represent the push-back force and vibrotactile feedback is rendered using a voice coil actuator to mimic the constant flow of air. Girbés et al. (2016) developed a system with haptic feedback on the throttle in order to assist bus drivers. Similarly, Hosseini et al. (2016) introduced an assistance system that provides haptic feedback at the steering wheel to avoid collision between vehicles. Though several works have been introduced to provide haptic feedback during safety drills, nevertheless, there is still a lack of research to provide impact feedback for earthquake safety drills. Existing works for earthquake safety exercises only depended on the visual and aural senses (Li et al. 2017; Gong et al. 2015; Liang et al. 2018). The addition of haptic feedback with vision and audition could greatly enhance earthquake safety drills.

2.3 Impact haptic feedback

Impact feedback is a kind of physical stimulus that emerges during a collision with an object. An impact signal represents a strong impulse response along with a short vibration signal. Existing works either considered an impulse signal (Park et al. 2019; Poorten and Yokokohji 2006) or Pulse-Width Modulation (PWM) signal (Handa et al. 2019) for impact feedback. On the other hand, several studies also attempted to develop actuators for impact rendering. For instance, Park et al. (2019) designed a multimodal actuator using a voice coil actuator and impact actuator for collision effects rendering. In their work, the impact actuator contained three solenoids and a permanent magnet. The impact feedback happens when the magnet collides. Poorten and Yokokohji (2006) developed a system that rendered the impulsive force during the collision in VR. The system is composed of three main components, i.e., force generator, coupling part, and momentum generator. In contrast, Handa et al. (2019) designed a ball-type haptic interface, which generates impact vibration using a vibro-transducer and four servo motors when the ball hits objects. In Pyo et al. (2015), the authors designed a haptic actuator using a permanent

magnet and a solenoid coil for impact feedback. Similarly, in Kim et al. (2018), an impact actuator is designed to render a planer two-DOF impact as well as vibrotactile stimuli. Lopes et al. (2015) designed a device to render the haptic feedback using a solenoid for hitting or being hit in the VR environment. In contrast, numerous event-based haptic feedback were provided in Hwang et al. (2004), Kuchenbecker et al. (2006), Okada et al. (2021). Hwang et al. (2004) designed a haptic interface for tapping on a virtual wall. Later on, a method for generating contact transients has been developed in Kuchenbecker et al. (2006), and the performance is compared by acceleration matching. Recently, in Okada et al. (2021), a passive-type haptic interface based on a DC motor’s damping brake was employed.

The above-discussed studies provided impact feedback with impulse or PWM signal and ignored the variational impact feedback rendering except (Park et al. 2019). In Park et al. (2019), impact feedback was provided to the user’s hand for the collision effect by using a short impulse signal. They rendered the ratio between force and mass as impact feedback and overlooked the velocity of the object before colliding. Furthermore, their approach cannot render the true impact feedback, since the true impact signal includes a small vibration signal along with the impulse signal.

3 System and rendering hardware

This paper presents a new and innovative system that provides realistic impact feedback of falling objects during an earthquake in a virtual reality environment, which can be utilized for earthquake simulation exercises. The overall system is briefly illustrated in Fig. 1. An earthquake is simulated in a virtual environment (VE) and a virtual human model is used to represent the user in the simulation. Various objects fall onto the user’s head in the VR environment during the earthquake. The collision is detected when virtual objects fall onto the simulated user’s head. Our data-driven impact feedback module receives the collision information,

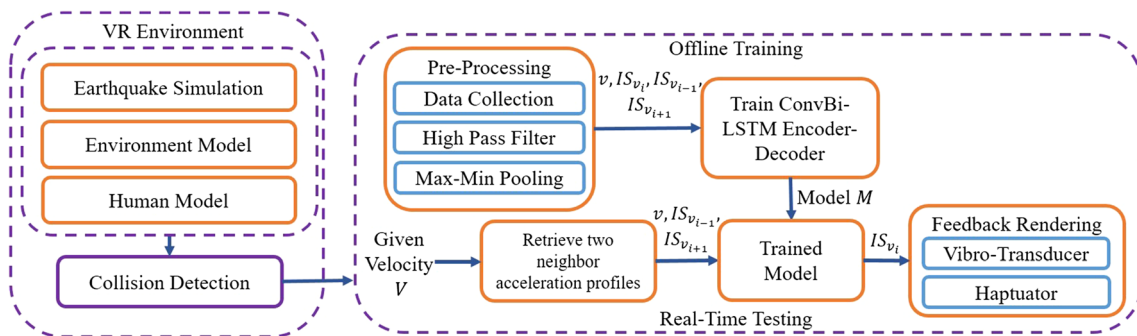


Fig. 1 An overview representing the overall flow of the proposed system

i.e., impact velocity and falling object from the VE, and estimates actual acceleration profile for the target velocity and object. This estimation is based on the deep neural network trained using actual acceleration signals.

3.1 Virtual environment modeling

To simulate a virtual earthquake, we build virtual environments in Unity 2019. The room and objects are represented as 3D meshes. For designing a realistic room, we arrange furniture and objects commonly available in a room. A human model is used to represent the user in the simulation. Furthermore, we also placed four virtual objects, (i.e. plastic frame, concrete brick, piece of wood beam and steel plate) that fall down onto the user's head during the earthquakes. Other furniture and objects shake during the simulated earthquakes in the VR environment.

We visually simulate an earthquake by shaking the virtual room based on the actual earthquake data. Similar to Li et al. (2017), we utilized the 1952 Kern County earthquake data of magnitude 7.3, which happened in Los Angeles. The shake disseminates from the floor to all the virtual objects placed in the room. If any virtual objects fall onto the head of the virtual human representing the user, our approach detects the collision employing Unity's physics engine. The total duration of the earthquake is set to 60 s. The scenes of the virtual environment used in our experiments are demonstrated in Fig. 2a.

3.2 Hardware setup

As a rendering actuator, building a sophisticated actuator for rendering of impact is out of scope of the paper. Instead, considering that the final stimulus due to impact is fast changing acceleration, we utilized commonly available actuators capable of generating rapid acceleration. Three options are chosen and compared later. These are Vibro-transducer (Vp408), Haptuator (MM1C, Tactile Labs), and push-pull solenoid (JF-0826B; input range ± 12 V). Vibro-Transducer has an input range of ± 5 V and frequency response between 20 and 15,000 Hz. The size of employed vibro-transducer is $17.2(H) \times 56(W) \times 56(D)$ mm. In contrast, haptuator has an input range of ± 5 V and frequency response between 30 and 800 Hz. The dimensions of haptuator are $23.95(L) \times 9.5(W) \times 9.5(H)$ mm. Vibro-Transducer and Haptuator both are voice coil actuators. They are mainly designed to provide vibrotactile feedback. The push-pull solenoid is a short-length linear motor capable of generating actual collision. The size is $22(H) \times 25(W) \times 26(L)$ mm along with a plunger diameter of 7.4 mm. For attaching them to user's head, we firmly attached the haptic actuators with a safety helmet as shown in Fig. 3. Users wear this helmet in order to feel the object falling impact feedback.



(a)



(b)

Fig. 2 **a** The scenes of the virtual environment used in our experiments and **b** The experimental setup used during the evaluation

While all the three actuators are used for the baseline physics-based approach, only the vibro-transducer and the haptuator are used for the data-driven impact feedback rendering since the push-pull solenoid is not controllable by an arbitrary waveform. Figure 4 illustrates the mechanism and processes of actuator controlling. Vibro-transducer and haptuator are controlled using a soundcard and amplifier, while push-pull solenoid is regulated using Arduino Uno.

4 Data-driven impact feedback

In general, data-driven rendering algorithm first collects physical signals related to the feedback during actual interaction, second, builds an input-output mapping database and interpolation scheme for missing data, and third, estimates proper signal during rendering based on the user's input and the mapping. The core part is step two: proper mapping and interpolation, which determine the performance of the algorithm. In literature, numerous methods are proposed, e.g., Culbertson et al. (2014), Yim et al. (2016), Abdulali and Jeon (2016), Abdulali et al. (2018), Osgouei et al. (2020), Shin and Choi (2020) where Radial Basis Functions Network (RBFN), Linear Predictive Coding (LPC), or Neural

Fig. 3 Proposed prototype for impact feedback rendering using different actuators, **a** Vibro-Transducer, **b** Haptuator, and **c** Push-Pull Solenoid

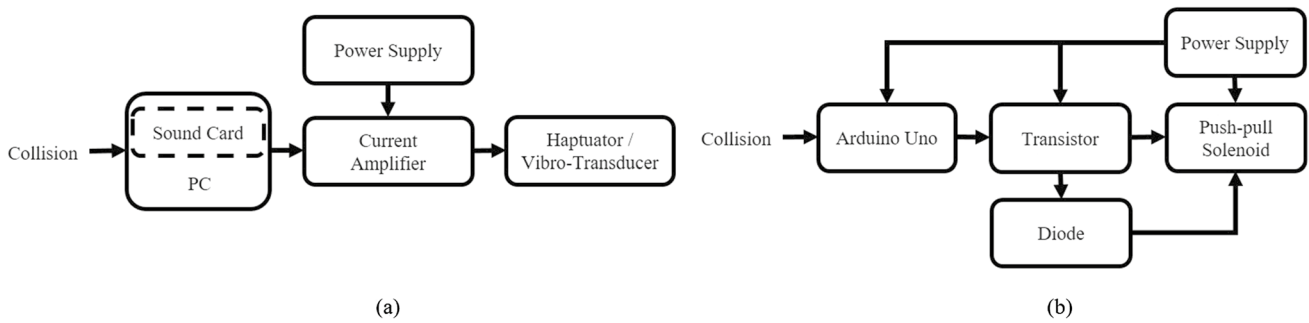
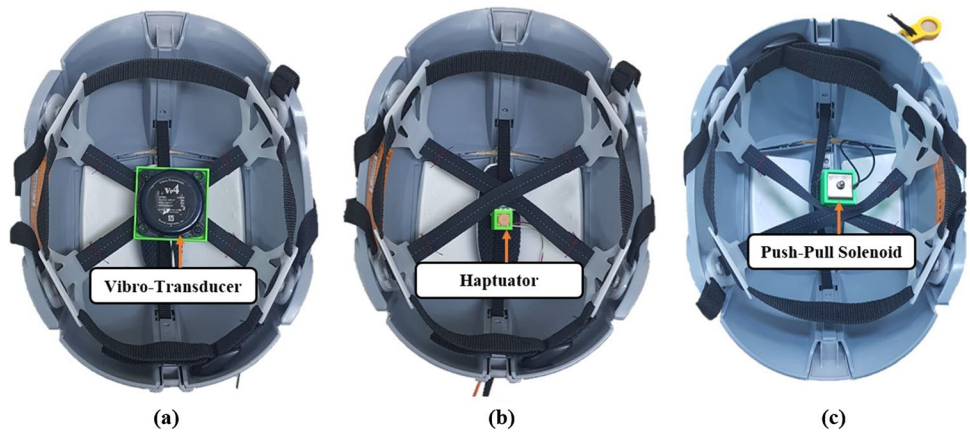


Fig. 4 Actuators controlling mechanism and process for **a** haptuator and vibro-transducer and **b** push-pull solenoid

Network (NN) are used for mapping the user’s interaction space to the force response.

The situation in our case is a bit different. The output impact acceleration signal does not depend on the user’s interaction but on the falling object and on the impact velocity of that object. Thus, for a certain object, the mapping should be done between the object’s velocity and the impact acceleration profile. However, it is difficult to map an entire impact acceleration profile with respect to a single velocity value. Another concern is that the object velocity is a continuous variable, and we only have a limited number of impact acceleration profile samples for particular velocities. The interpolated acceleration signals for remaining velocity values (i.e., for which we did not record the impact acceleration profile) are needed. The above concerns motivated us to formulate this problem as a multivariate sequential time series prediction problem, where we predict a new impact acceleration profile for a given velocity by taking two neighboring impact acceleration responses for lower and higher impact velocities from database as inputs. More specifically, given a multivariate time series $x = \{IS_{v_{i-1}}, IS_{v_{i+1}}\} \in \mathbb{R}^n$, where $IS_{v_{i-1}}$ and $IS_{v_{i+1}}$ are the two neighboring impact acceleration profiles (i.e., one with an immediate lower impact velocity and one with an immediate higher impact velocity from database) ($n = 2$) with length l and our goal is

to predict the 1D impact acceleration profile $y = \{IS_{v_i}\}$ of length l for the given velocity v . To this end, we propose a deep Convolutional Bidirectional Long Short-Term Memory (ConvBiLSTM) encoder–decoder framework to interpolate the impact acceleration profile for a given object velocity. Therefore, the input to the model consists of given velocity (v), two neighboring impact acceleration profiles ($IS_{v_{i-1}}$ and $IS_{v_{i+1}}$), while the output is impact acceleration profile IS_{v_i} corresponding to the given velocity. Finally, the interpolated acceleration profile is rendered as impact feedback. This section first introduces the data-collection setup, and then moves to the deep-learning-based interpolation algorithm and rendering procedure.

4.1 Data collection setup

We build a custom data-collection setup where real objects are dropped on the helmet with different velocities to capture the acceleration signals due to the collision. For acceleration data capturing, a GY-61 accelerometer is attached to the inside of a safety helmet. Data are transmitted to the PC through a data acquisition card (DAQ NI-USB 6009). Impact acceleration data are captured by dropping a real object onto the helmet. To control the impact velocity at the moment of contact, we changed the initial height of the object. In order

to ensure that the object is dropped from exactly same place with same orientation, a rack capable of changing height is used for the positioning of the object before dropping, while the helmet is placed on the floor. Note that in our work, the effect of air drag is insignificant due to the following two reasons. Firstly, in our situation, the velocity of the object is very slow (2–5 m/s) and the effect of air drag can be considered negligible. Secondly, the four objects we used are quite heavy objects (from 35 to 200 g). Thus, the effect of the air drag on the velocity reduction is relatively insignificant. Therefore, for the sake of simplicity, we assumed that the velocities of the different objects are equal.

4.2 Data mapping and interpolation using deep ConvBiLSTM encoder–decoder

The collected acceleration profile can be directly used to generate impact feedback during rendering if the impact velocity of a virtual object is exactly same as the collected velocity. However, in most cases, the velocity does not hit the same values, and thus we employ the following interpolation algorithm to estimate the acceleration profile. The aim is to synthesize the acceleration signal for an arbitrary impact velocity even that was not trained by the proposed model.

In general, the interpolation of time-series high-frequency acceleration profiles is done in frequency domain (Abdulali et al. 2020), but in this case, exact phase information and features of the signal would be lost. This is in particular a critical problem in the impulse acceleration case. In opposite to the prolonged high-frequency acceleration profile, e.g., scratching a rough surface, where perceptually important information is mainly embedded in the frequency and amplitude of the signal, impulse acceleration possesses its perceptually significant features in the shape and timing of the signal. Thus, we proposed new signal interpolation scheme using a deep learning

technique, which keeps phase and actual shape information of the acceleration profile.

The Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM)-based deep learning approaches have become very popular for mapping sequential information. However, their performance deteriorates when the sequence length increases. Recently, to enhance the performance, a simple LSTM Autoencoder model was proposed in Srivastava et al. (2015), which includes one encoder LSTM and one decoder LSTM. Moreover, an LSTM-based stacked autoencoder was developed in Sagheer and Kotb (2019) for multivariate time series forecasting. Extending this work, we propose a deep Convolutional Bidirectional Long Short-Term Memory (ConvBi-LSTM) encoder–decoder model to effectively predict the impact signal for a given object velocity. Compared to simple (Joolekha and Jeon 2022), the ConvBi-LSTM employs convolutional kernels in the input-to-state and state-to-state transitions of the BiLSTM. Therefore, the proposed architecture preserves both spatial and temporal information more effectively. Figure 5 illustrates architecture of our deep ConvBi-LSTM encoder–decoder framework.

The proposed ConvBi-LSTM encoder–decoder is mainly composed of encoding and decoding layers. In the encoding layer, three ConvBiLSTM layers are employed with 256, 128, and 64 filters, respectively, with kernel sizes of 1×3 . The convolutional operation is responsible for obtaining the spatial information, whereas the BiLSTM captures the temporal dynamics. To address the overfitting issue, each ConvBiLSTM layer is followed by a dropout layer. The basic update equations of ConvBiLSTM at time t are as follows:

$$i_t = \sigma(W^i * x_t + R^i * h_{t-1} + U^i \circ c_{t-1} + b^i) \quad (1)$$

$$f_t = \sigma(W^f * x_t + R^f * h_{t-1} + U^f \circ c_{t-1} + b^f) \quad (2)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W^c * x_t + R^c * h_{t-1} + b^c) \quad (3)$$

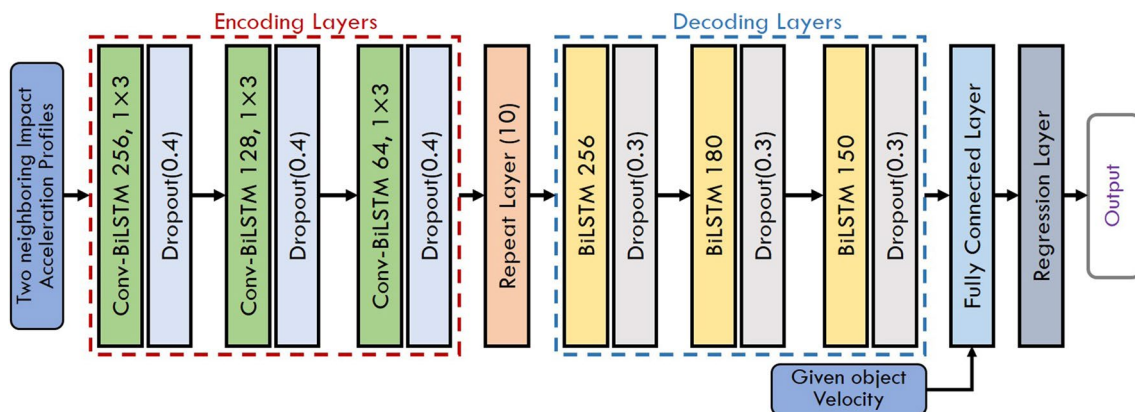


Fig. 5 The architecture of deep ConvBi-LSTM encoder–decoder model

$$o_t = \sigma(W^o * x_t + R^o * h_{t-1} + U^o \circ c_t + b^o) \tag{4}$$

$$h_f = o_f \circ \tanh(c_f) \tag{5}$$

$$h_b = o_b \circ \tanh(c_b) \tag{6}$$

$$h_t = (h_f, h_b) \tag{7}$$

where i_t, f_t , and o_t denotes the input, forget and output gates, respectively. c_t , σ and \tanh represents the cell state, logistic function and hyperbolic tangent function, respectively. Whereas, W^i, W^f, W^c , and W^o denotes the input to state filters and R^i, R^f, R^c , and R^o represents state to state filters. $U^* \circ c_{t-1}$ indicates Hadamard terms. b^i, b^f, b^c and b^o are the bias parameters. In the BiLSTM, the forward and backward layers are estimated by h_f and h_b , respectively and h_t represents the hidden state. The network takes multivariate time-series sequence $x = (IS_{v_{i-1}}, IS_{v_{i+1}})$ as input, therefore, at each time step t , the input is x_t .

A repeat layer is used to decode the feature maps acquired from the encoder layers. This layer repeats the final output vector from the encoding layer in a shape that is a constant input to each time step of the decoder. The decoding layer can then reconstruct the original input sequence. Afterward, three BiLSTM layers are employed in the decoding layer with 256, 180, and 150 hidden nodes, respectively. In a hierarchical manner, the output of the first BiLSTM layer is fed into the second BiLSTM layer and the output of the second BiLSTM layer is fed into the third BiLSTM layer. Therefore, the decoder layer can incorporate the output features from the encoding layer, which enhances the predictive model efficiency by facilitating representation learning on each layer. Finally, the model ends with a fully connected layer and a regression layer. Since the object velocity is a single value, therefore, it is directly fed into the fully connected layer. In our work, the root-mean-square error (RMSE) is used as the loss function.

As a result, the trained network is capable of predicting a full time-series acceleration profile that is the result of weighted interpolation between two given time-series acceleration profiles and one target velocity. The features and characteristics of the data used for the training of the network are stored in the model, and the interpolation tries to depict these features to make a proper estimate.

4.3 Data driven modeling and rendering

Overall training and rendering procedures are depicted in Algorithm 1. For training the network, we collected data using the data collection setup. For each dropping, one impulse acceleration profile is captured at 1 kHz. One impulse lasts about 200–400 ms, thus 500 ms was selected

as the length of one signal, yielding 500 data points in an acceleration profile. Twelve different velocities are used for data collection, and for each velocity, ten drops are used.

Once the acceleration data is collected, another part essential for the modeling and rendering is the filtering of the signal. Here, we employed a high-pass filter at 20 Hz to the acceleration signals for eliminating gravity and the static acceleration reading. Haptics researchers mostly use a single-axis actuator to render the feedback since humans cannot easily perceive the direction of vibrations, and using this single-axis actuator significantly reduces the system cost and complexity (Park and Kuchenbecker 2019). Therefore, many approaches were presented in Park and Kuchenbecker (2019), to convert the 3D acceleration signals into a 1D signal. For simplicity without losing perceptual accuracy, we convert the 3D acceleration signals into a 1D signal by employing a new approach, namely, max–min extraction approach. The proposed max–min extraction approach for synthesizing 1D acceleration from the 3-axis acceleration signal is presented in Eq. (8). A_x, A_y and A_z denotes the values of three-axis acceleration signals. The max function returns the maximum value, whereas the min function returns the minimum value among A_x, A_y and A_z . The main advantage of the max–min extraction approach is that it is simple and fast. The max–min extraction and filtering is applied to all acceleration signals before training and rendering. For each velocity sample, the representative acceleration profile from the collected ten impact acceleration profiles (see Fig. 6a) is predetermined in advance. The selection is done manually by carefully inspecting the acceleration profile and choosing one that has average characteristics among the collected acceleration profiles. This process is done only once. Note that, we overlook simple mathematical operations (e.g., averaging) to intake the original signal characteristics. Figure 6b presents different impact profiles for piece of wood beam with different velocities.

$$IS_{v_i} = \begin{cases} \max(A_{x_i}, A_{y_i}, A_{z_i}); & \text{if } A_{x_i} \geq 0 \text{ or } A_{y_i} \geq 0 \text{ or } A_{z_i} \geq 0 \\ \min(A_{x_i}, A_{y_i}, A_{z_i}); & \text{Otherwise} \end{cases} \tag{8}$$

After collecting and pre-processing the data, the deep network is trained. Single network is trained using the whole data. Thus, the trained network now possesses the features of the acceleration profiles associated with object and velocity. During rendering, the Unity simulation engine gives a collision event, which comes along with the object ID and the impact velocity. Given the target velocity v , our algorithm finds out two neighboring acceleration profiles $IS_{v_{i-1}}$ and $IS_{v_{i+1}}$, one with an immediate lower impact velocity v_{i-1} and one with an immediate higher impact velocity v_{i+1} from the database containing recorded signals.

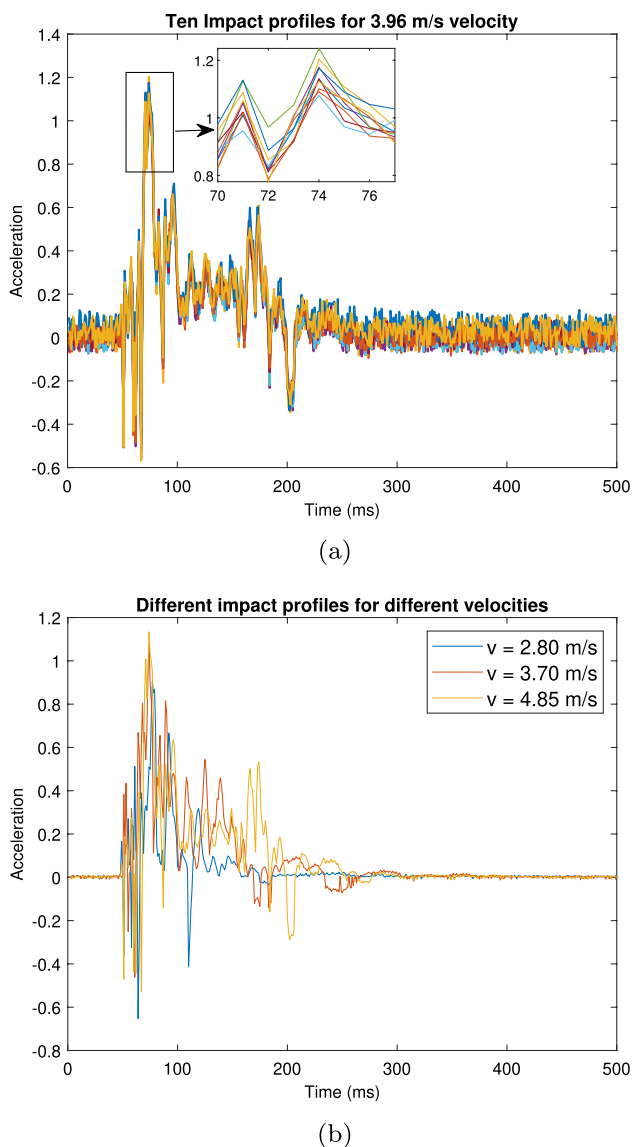


Fig. 6 **a** Ten impact profiles for the piece of wood beam object with 3.96 m/s velocity and **b** Different impact profiles for wood with different velocities

Then, the network takes the given velocity v and the two neighboring acceleration profiles $IS_{v_{i-1}}$ and $IS_{v_{i+1}}$ for the given velocity as input and returns an interpolated full acceleration profile for the target velocity. Finally, the estimated acceleration profile is passed to the actuator controller for the rendering.

5 Evaluation

This section presents the experimental evaluation of the proposed approach. The evaluation is divided into three subsections: numerical evaluation of the proposed network,

subjective evaluation along with perceptual study and performance of max–min extraction approach.

5.1 Numerical evaluation

In this section, we numerically evaluate the data-driven impact feedback algorithm. The data-driven algorithm synthesizes acceleration signals for an arbitrary impact velocity even that was not trained by the proposed ConvBi-LSTM encoder–decoder. The present numerical evaluation is designed to test how close the synthesized acceleration profiles coincide with the ground-truth measured acceleration signals at the target impact velocity, which is not used during network training.

Dataset First, for each object, acceleration profiles IS_{v_i} were captured at twelve different velocities (i.e., $v = 2.42, 2.80, 3.13, 3.43, 3.70, 3.96, 4.20, 4.43, 4.65, 4.85, 5.05,$ and 5.24 m/s). Therefore, we have total 12 acceleration profiles for twelve different velocities. Then, we trained the model for five velocities, $v = 2.80, 3.13, 3.43, 4.85,$ and 5.05 m/s. At that time, for each velocity v along with its acceleration profile $y = \{IS_{v_i}\}$ (i.e., l is the length of the acceleration profile), we also taken the two neighboring impact acceleration profiles $IS_{v_{i-1}}$ and $IS_{v_{i+1}}$ as input. For instance, if we need to train the model for velocity $v = 3.13$ m/s, then the proposed model takes its impact acceleration profiles $IS_{v_{3.13}}$ along with two neighboring impact acceleration profiles $IS_{v_{2.80}}$ and $IS_{v_{3.43}}$ (i.e., one with an immediate lower impact velocity $v_{2.80}$ and one with an immediate higher impact velocity $v_{3.43}$) as input. Then, all these inputs were used to train the model. For each object, to test the performance of the model, $v = 3.96$ m/s and 4.43 m/s were selected. Therefore, total 20 samples (5 velocities \times 4 objects) were used for training, and 8 samples (2 velocities \times 4 objects) were used to test the model. Furthermore, each sample has $l = 500$ data points as mentioned in Sect. 4.3.

The evaluation was done for four real objects. It is well known that during an earthquake different types of objects may fall down, e.g., broken beams, bricks from the wall, parts of a ceiling, lighting bulbs, plastic frames of the light, fans, and so on. To cover the variety of the object, this study utilized four different objects with different materials, i.e., a plastic frame, a concrete brick, a wood beam, and a steel plate (see Fig. 7). Their parameters are described in Table 1.

Hyperparameters To train the model, ADAM optimization approach is applied with batch size of 8, a momentum of 0.9 and a learning rate of 0.0001. To prevent overfitting, dropout is set to 0.4 in encoding layers and 0.3 in decoding layers, respectively. All of the hyperparameters detailed above are determined empirically.

Ablation Study Figure 8 shows the examples of the estimated and collected acceleration profile for the velocity 3.96 m/s. In most cases, the estimated signals well coincide with



Fig. 7 Four objects: Plastic frame, Concrete type broken brick, Wood type broken beam, and Steel type ceiling part used for the experiments

Table 1 Parameters for four objects

Objects	Mass (g)	Dimensions (width × height × depth) (mm)
Plastic frame	35	53.4 × 81.9 × 44.7
Concrete brick	69	36.5 × 48.3 × 14.6
Piece of wood beam	150	58.2 × 98.3 × 35.6
Steel plate	200	100 × 100 × 4.8

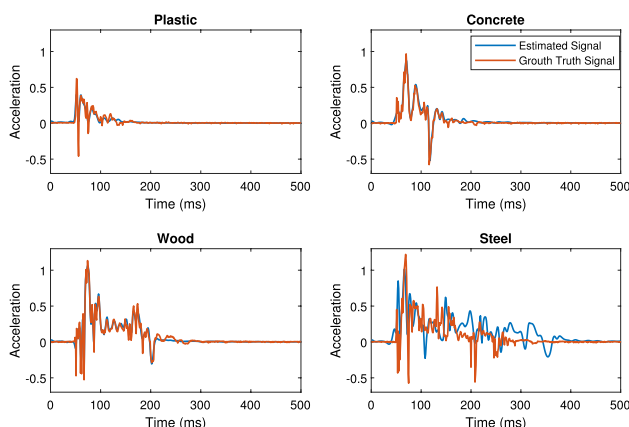


Fig. 8 Examples acceleration profiles: measured versus estimated using our approach for velocity 3.96 m/s

the measured signals. This is a very promising result since in signal estimation tasks such an exact-match in time-domain for high frequency signal is a quite challenging task.

In order to objectively compare, we calculated the Root Mean Square Error (RMSE) as an error metric. Table 2 reports the RMS error for four objects and two velocity values that were selected as the test data. The plastic, concrete, and wooden objects showed lower RMS errors, while the steel objects showed higher RMS errors. Our speculation on this result is that the shape of the steel object is a thin plate, while other objects are either cube or spherical shape. While falling, thin plate object, i.e., the steel object, the trajectory varies from time to time, resulting in the slight variations in the impact orientation

Table 2 Root mean square error for the estimation of the acceleration profile

Objects	RMSE for velocity 3.96 m/s	RMSE for velocity 4.43 m/s
Plastic frame	0.033	0.025
Concrete brick	0.027	0.032
Piece of wood beam	0.041	0.040
Steel plate	0.162	0.114

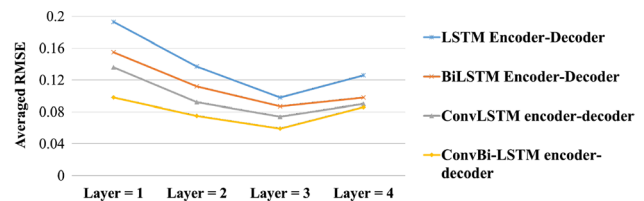


Fig. 9 Averaged RMSE comparison between different models with varying the number of encoder–decoder layers

and location in each trial. This uncontrollable factor is one of the causes of the higher RMSE in the steel object case. In the domain of haptics, an RMS error in the range from 0.025 to 0.162 cm/s² for the vibrotactile signal was regarded as insignificant (Abdulali et al. 2020; Romano et al. 2010). Nevertheless, due to differences in the situation, i.e., vibrotactile due to texture vs. impact, it is still unclear how significant this error indeed is in perception. The following section examines the effect of the errors through a series of subjective and perceptual experiments.

To make sure whether our proposed interpolation scheme is optimum for this task, we also compare RMSE from our proposed network (ConvBi-LSTM encoder–decoder) with existing state-of-the-art models, i.e., weighted averaging scheme-based linear interpolation(Hassan et al. 2020), Single-layer LSTM, Two-layer BiLSTM, LSTM-SAE (Sagheer and Kotb 2019) (LSTM autoencoder), Bi-LSTM encoder–decoder and ConvLSTM encoder–decoder). Note that the ‘weighted averaging scheme’ averages two neighboring impact acceleration data based on weight in the time-domain. Both the single-layer LSTM and two-layer BiLSTM include 200 hidden nodes for interpolating the impact signal. Figure 9 presents the averaged RMSE comparison between different models with varying the number of encoder–decoder layers. Note that, the averaged RMSE is obtained by taking the average of the RMSE values for four objects and two testing velocities data. Table 3 reports the averaged RMSE comparison with different methods. The results show that our model exhibits the lowest RMSE, ensuring the relative appropriateness of our model.

5.2 Subjective evaluation and perceptual study

A total of two user studies were carried out to assess the performance and efficacy of the proposed impact rendering method. The first experiment is designed to evaluate the overall perceived realism of the feedback in comparison with the baseline physics-based approach. The second experiment goes further; the exact similarity between the virtual and real feedback is examined along with the virtual feedback matching. All the human experiments performed in this work were approved by the Institutional Review Board at the authors' institution (KHGIRB-21-321).

5.2.1 User study 1: overall realism

For comparison, we implemented a conventional physics-based approach for impact rendering. The approach simulates haptic feedback based on physics equations. In literature, physics-based approaches are quite common in haptic rendering, e.g., Xu et al. (2019), Park et al. (2019), Chan and Choi (2009), Park and Choi (2017), but no prior work exactly fits to our case. Thus, we implement the algorithm as follows. In general, when an object falls onto the user's head, the kinetic energy is transformed into vibration generating impact feedback. Assuming the duration of the energy transformation is fixed for a certain object, the amplitude of impact is proportionally related to the kinetic energy. This is described by

$$KE = \frac{1}{2} \times m \times v^2, \quad (9)$$

$$\text{Amplitude} = \frac{KE}{KE_{\max}}, \quad (10)$$

$$f(t) = \begin{cases} \text{Amplitude}; & \text{if } 0 < t < b \\ 0; & \text{otherwise} \end{cases} \quad (11)$$

where m is a mass of the falling object, and KE_{\max} is the maximum amplitude that a certain actuator can produce. When the collision is detected, the velocity and eventually the kinetic energy of the object is calculated by the Eq. (9).

The square-wave function as presented in Eq. (10) is used to generate the impact feedback, which is then provided using the three different kinds of actuators, i.e., vibro-transducer, haptuator, and push–pull solenoid.

Participants A total of 15 participants (10 males and 5 females) took part in this user study. Their mean age was 29.5 years (ranges from 24 to 35 years). They were informed of the experimental procedure beforehand. Four participants had a moderate experience of using haptic devices, while the others were naive. No participant reported any disabilities that would restrict the experimental procedure.

Experimental Conditions In this study, a total of four virtual objects were used, with the same parameters as described in Table 1. Two rendering algorithms, i.e., physics-based and data-driven algorithm, are used for the comparison. All three helmets were used for the physics-based algorithm, while the push–pull solenoid-based helmet was not used for the data-driven approach due to the inability to control the push–pull actuator with an arbitrary waveform profile. Thus, the total number of conditions was 12 (4 objects \times 3 helmets) for the physics approach and 8 for the data-driven approach (4 objects \times 2 helmets), yielding 20 conditions. Note that in the data-driven approach, for velocity 3.96 m/s and 4.43 m/s, we rendered the estimated acceleration profiles rather than rendering the captured acceleration profile. These velocities were selected as target velocities to test the performance of the model. For the other cases, we rendered the captured acceleration profiles.

Procedure The experimental session consisted of training and main sessions. During the training session, at first, a monitor screen displaying the virtual scene is presented to the participants, where participants were introduced to the virtual objects and falling of these objects onto the head of the human model in VR. Then, impact feedback is explained to the user for the falling objects on the head. Furthermore, the output impact responses for the physics-based and data-driven approach were also illustrated to the participants. On average, the training session lasted for 10–15 min. Afterward, the participant took a break for 5 min.

In the main session, participants sat in a chair in front of a computer monitor while wearing an Oculus Rift and one of the helmet as shown in Fig. 2b. In the virtual reality

Table 3 Averaged RMSE comparison between the proposed ConvBi-LSTM encoder–decoder and existing state-of-the-arts

Methods	Averaged RMSE
Weighted Averaging Scheme based linear interpolation (Hassan et al. 2020)	0.297
Single-layer LSTM	0.196
Two-layer BiLSTM	0.148
LSTM-based stacked autoencoder (Sagheer and Kotb 2019)	0.098
Bi-LSTM encoder–decoder	0.087
ConvLSTM encoder–decoder	0.074
ConvBi-LSTM encoder–decoder	0.059

environment, the user could see the objects falling on their head. Furthermore, participants wore headphones that played earthquake sound. For each participant, the main session contained 12 + 8 conditions. For each condition, 7 velocities were used. As a result, 84 + 56 trials were done for each participant. In each trial, impact feedback from four virtual objects was rendered to the user’s head, synchronized with the graphics and sound. After each condition, the participant was asked to rate the overall feedback fidelity by answering two questions: (1) how much realistic the feedback was? (Realism) and (2) was the feedback unnatural? (Unnaturalness). The rating was given in the form of 0–100 continuous scale. The entire experiment took about 2 h on average for a participant.

Results The mean scores for the questionnaires are plotted with standard errors in Fig. 10. After confirming that the assumptions of ANOVA were met, we carried three-way repeated measures ANOVA on each question applying three independent variables, i.e., actuator, object and rendering method. Note that the push–pull solenoid actuator result was not considered, because it was not used for the data-driven approach, and even in the physics-based approach, it is outperformed by the other actuators. The rendering method had a significant effect on all the questions; realism ($p = 0.0001$) and unnaturalness ($p = 0.0002$). Similarly, object and actuator also had a statistically significant effect on all the questions; (realism, $p = 0.0051$ and unnaturalness, $p = 0.0015$) and (realism, $p = 0.002$ and unnaturalness, $p = 0.0014$), respectively. For realism questionnaire, the interaction term (actuator \times object) and (object \times rendering method) were not statistically significant ($p > 0.05$). However, the interaction term (actuator \times rendering method) were statistically significant ($p = 0.0137$). On the other hand, for the unnaturalness questionnaire, all the interaction terms were not statistically significant ($p > 0.05$). Later on, Bonferroni-Holm analysis was utilized for pair-wise post hoc tests. For both the questionnaires, Bonferroni-Holm analysis demonstrates that except (actuator \times rendering method), all the pairs scores were not significantly different. For the realism questionnaire, participants preferred the vibro-transducer based helmet with the data-driven approach. In contrast, the physics-based approach along with the push–pull solenoid was rated lower. For both rendering methods, the participants rated the wooden object as the best. Besides, actuators along with the rendering approaches were also inspected in the form of box plots in Fig. 11 a, b with regard to realism and unnaturalness, respectively. For both the rendering methods as well as for the questionnaires, vibro-transducer shows superior performance.

Discussion While both approaches are positively scored (for the realism questionnaire, and both actuators, the median is 70 for the physics and 85 for the data-driven approach), the proposed data-driven rendering system

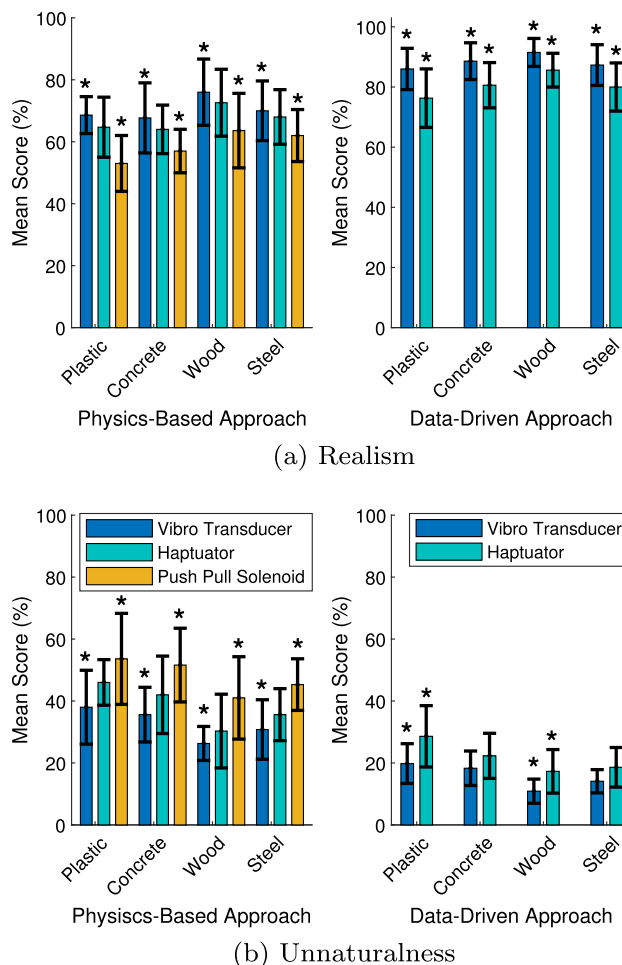


Fig. 10 Mean scores for the **a** realism and **b** unnaturalness. The error bars show the standard deviation. The * shows statistically significant difference with $p < 0.05$. Note that, **a**, **b** legends are same

received significantly better ratings for both actuators. The physics-based system, being driven by parametric equations, offers a simple and fast algorithm for rendering the impact of falling objects with easier implementation and less computation power. On the other hand, it is clear that the data-driven system provides richer haptic feedback.

5.2.2 User study 2: perceptual study

Two experiments are conducted in this user study; one is virtual feedback matching and the other one is the similarity of the haptic feedback itself. One of the straightforward ways of assessing the haptic accuracy of virtual feedback is to directly compare the virtual feedback with corresponding real feedback. This is performed in both these experiments.

Participants A subset of 10 participants from the previous experiment took part in this experiment. To avoid learning effects, a 6-week washout period (Bortone et al. 2020) was

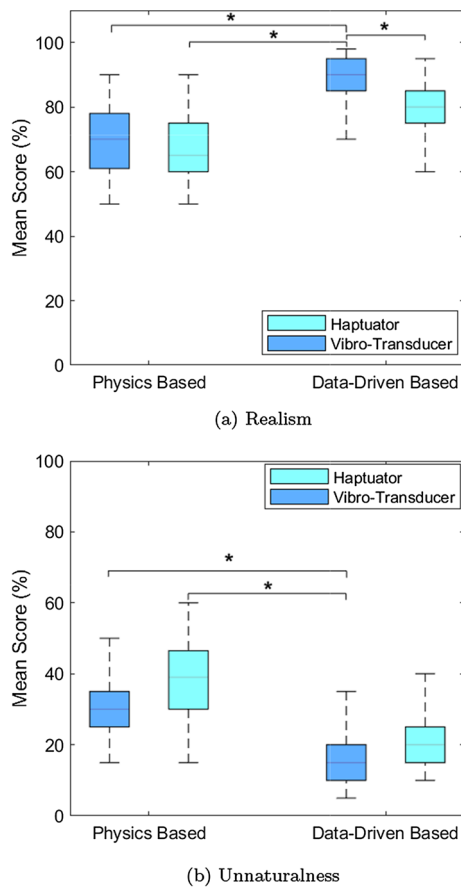


Fig. 11 Comparison of different actuators in the form of box plots. Asterisks shows statistically significant difference with $p < 0.05$

included between the two user studies, providing a significant time delay between participants involvement.

Experimental Design This experiment is divided into two parts: feedback matching experiment and similarity rating experiment. For each participant, the virtual impact feedback matching study was conducted first, followed by the similarity evaluation study. In the feedback matching study, participants were presented one virtual stimulus and asked to find the best-matching real stimulus as well as to find the best-matching virtual stimulus under a given real one. If a participant properly matches the virtual and real feedback, then it is counted as 1, otherwise 0. For the similarity rating study, they were asked to rate the similarity between virtual stimuli and corresponding real stimuli. In both studies, physics-based and data-driven algorithms were used for the virtual feedback, while only the vibro-transducer was used as the actuator, since it showed the best performance in user study 1. Velocities used in these experiments were 3.70 and 3.96 m/s.

While our rendering system created virtual stimuli, real stimuli were presented by the setup shown in Fig. 12. The object was carefully placed directly above the head with the



Fig. 12 Perceptual experiment. The participant compares the virtually rendered impact feedback and feedback from real objects falling. A safe guard was attached to the outside part of the helmet for safety reasons

set height (based on the velocity) and dropped. The object fell and hit the helmet, generating feedback. The object may bounce after hitting the helmet, which may cause injury to the participant. In order to prevent injury, safety guards are attached to the outside of the helmet, as shown in Fig. 12. The safety guards are two sponge foam plates covering the outside of the helmet to prevent the bounced object from hitting the participant's other body parts. The sponge guards do not affect the feedback delivered to the participant.

The feedback matching experiment consisted of three steps. First, a participant was presented with one virtual feedback (target) and then asked to find best-matching real feedback among the four real stimuli (comparisons). The second step is the other way around, finding the best-matching virtual one among the four virtual ones (comparisons) under a given one real stimulus. The last step was the matching between real and real pairs, acting as a reference. The participant was presented one target real stimulus and instructed to find one best-matching stimulus among the four real stimuli. This last step is included for testing how similar the discriminability of the virtual feedback is compared to that of real feedback.

For the similarity evaluation, there were total ten stimuli pairs (conditions). The first four are real–virtual pairs; for each real object, corresponding virtual feedback was paired, yielding four pairs, i.e., VP–RP, VC–RC, VW–RW, and VS–RS. These were repeated for both rendering methods. The next four pairs are for real–real comparisons. For each object, real stimuli are presented twice, yielding four real–real comparison, i.e., RP–RP, RC–RC, RW–RW, and RS–RS. These conditions act as the upper reference for the similarity ratings. Note that, even though exactly the same two physical stimuli are presented to a human, a similarity score does not usually reach 100 due to error in the human perception system. Thus, measuring the upper bound can

be helpful for the interpretation of the result. The last two pairs are for the lower bound; completely different two real objects, RP–RS, as well as two virtual feedback, VP–VS, were used for the comparison, yielding a lower reference score.

Procedure Before the beginning of this experiment, written and verbal instructions were provided to the users describing the procedure. It was made sure that users perfectly learned the process by asking them to repeat what they had understood.

The experimental session is divided into training and testing sessions. During the training session, participants were introduced to the environment, the objects, their feedback, and their tasks. The participant was informed that, in this experiment, he/she will feel both real and virtual impact feedback of the falling object’s on the head through different methods. Afterward, the real objects feedback were given to the participant. On average, the training session lasted for 15 min. Then, the participant took a break for 5 min.

In the main session, a participant sat in a chair. Auditory and visual stimuli were blocked by headphones with white noise and blindfolds, respectively. He/she first conducted the feedback matching experiment. As mentioned, the participant was first given one of the virtual feedback and asked to find the best-matching real feedback among the four real stimuli given during the training session. This process is repeated for the four virtual feedback, and this ends step one. During the matching trials, he/she was allowed to feel the comparison stimuli again and again if wanted. In step two, one real feedback was presented and asked to find the best-matching virtual feedback among the four virtual stimuli given in step one. This process is also repeated for the four real feedback. Finally, four real–real pairs feedback matching were done. The presenting order of the feedback was randomized.

In the similarity rating experiment, for each participant ten pairs (conditions) are presented. For each condition, 2 impact velocities are used in 2 repetitions, yielding 4 repetitions. The presenting order of the conditions were randomized, and within a condition, the order of each stimulus in a pair was randomized. In each trial, the participant was asked to rate the overall haptic similarity on a zero to 100 scale between two stimuli. Zero represents that the two feedback felt completely different, while 100 means the two stimuli were perceptually identical. The entire user study took about 60 min on average for a participant.

Results The experimental results for the feedback matching are presented in the form of confusion matrices in Fig. 13. The recognition rates were 32.5% with the physics-based approach and 45% with the data-driven approach for real–virtual matching. For virtual–real matching, the recognition rates were 35% and 47.5% with the physics-based and data-driven approaches, respectively. The recognition rate of

real–real matching was 70% with 12 mismatches out of 40. It is observed that the participants often become confused between the concrete and steel object feedback. Figure 14a, b shows the overall trend of the recognition rate for different conditions.

The experimental result for similarity evaluation is demonstrated in Fig. 15. The averaged similarity scores for the four real–real comparisons are used as the upper reference, whereas the average score across the three lower reference conditions is used for the lower bound. After confirming that the assumptions of ANOVA were met, we carried out a two-way repeated-measures ANOVA with object and rendering method as the two independent factors for the similarity rating experiment. The outcomes revealed that the similarity score was significantly different across objects ($p = 0.0027$) and between two rendering approaches ($p = 0.0016$). However, their interaction term was not statistically significant ($p = 0.6725$). Later on, for pair-wise posthoc tests, the Bonferroni-Holm analysis was employed. This post hoc test shows that all the pairs scores were not significantly different ($p > 0.05$). Note that, real feedback ratings were not included in the two-way ANOVA since they are used to produce the upper and lower reference lines. On average, the highest-rated pair of real–virtual objects was VW–RW, having a score of 69.5% for the data-driven approach, whereas the lowest was 44.5% for VC–RC during the physics-based

Step - 1					Step - 2				
	VP	VC	VW	VS		RP	RC	RW	RS
RP	5	2	1	2	VP	4	3	1	2
RC	1	2	2	5	VC	1	3	1	5
RW	1	3	3	3	VW	1	3	4	2
RS	2	3	2	3	VS	1	3	3	3

(a) Physics Based

Step - 1					Step - 2				
	VP	VC	VW	VS		RP	RC	RW	RS
RP	5	3	0	2	VP	6	3	0	1
RC	2	3	2	3	VC	1	4	2	3
RW	0	2	6	2	VW	0	2	6	2
RS	1	3	2	4	VS	1	4	2	3

(b) Data Driven Based

	RP	RC	RW	RS
RP	8	1	0	1
RC	0	6	1	3
RW	0	1	7	2
RS	0	2	1	7

(c) Real feedback guessing

Fig. 13 a, b Confusion matrices of virtual impact feedback matching by two rendering methods. c Confusion matrices of real object feedback matching

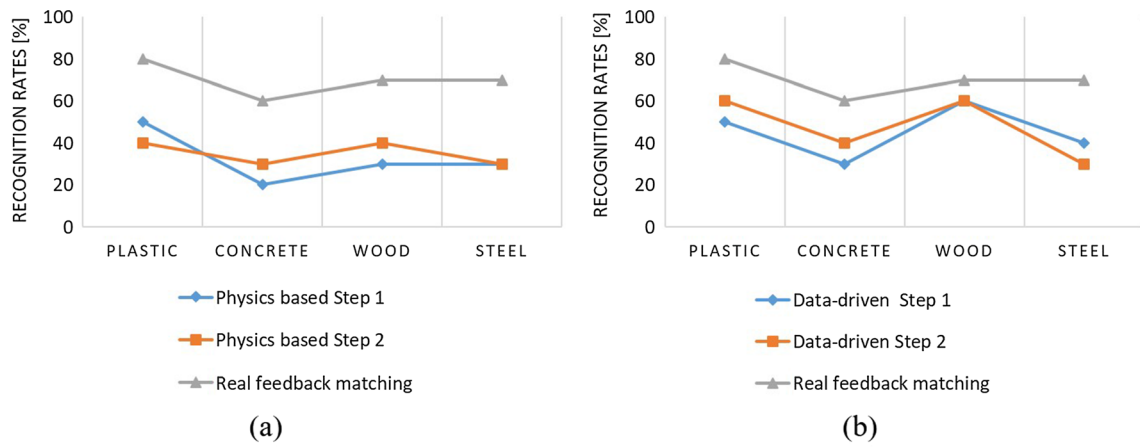


Fig. 14 Recognition rate comparison between real and virtual feedback matching with **a** Physics based **b** Data-driven approach

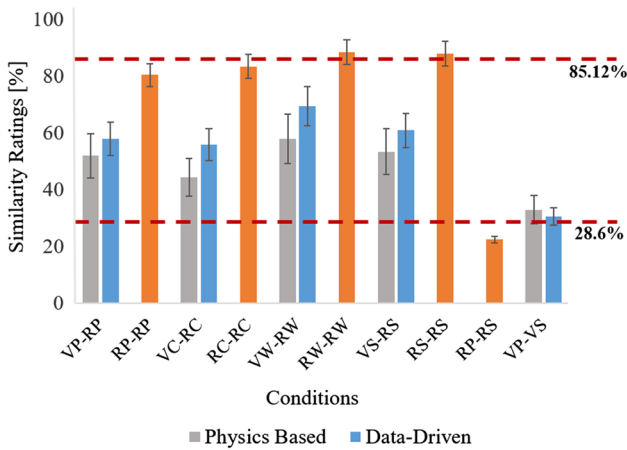


Fig. 15 Average scores from the similarity evaluation. The error bars show the standard deviation. Note that, V represents virtual, R represents real, P denotes plastic, C represents concrete, W denotes wood, and S represents steel

approach. The scores of real–virtual pairs remained consistent, ranging from 44.5% to 58% for the physics-based approach and 56% to 69.5% for the data-driven approach. On average, the similarity of the feedback from the data-driven approach reaches 61.125%, whereas 52% for the physics-based approach.

Discussion In the feedback matching experiment, it is observed that the overall recognition rate of the virtual matching conditions is inferior to that of the real matching condition. This trend is stronger in physics-based method. However, in the data-driven method, the recognition rates of virtual matching conditions are quite comparable to those of real matching conditions (see Fig. 14b). This indicates that at least for the data-driven method, perceptual features used in discriminating different kinds of impact are captured well in virtual feedback with reasonable quality. This is further

supported by the fact that the trend of the virtual matching recognition rate of the four different objects exactly follows that of real feedback matching, as presented in Fig. 14a, b. For instance, we could observe that the discriminability of plastic and wood type objects is higher than the concrete and steel type objects for both virtual feedback and real feedback. From the confusion matrices (see Fig. 13a, b) it is evident that, there was a mix-up between concrete and steel feedback, which contributed substantially to reduce the recognition rates for both virtual and real feedback matching.

In the similarity evaluation study, participants rated the virtual-real pairs positively, although the scores were still below the upper reference line. Among all the virtual feedback, concrete type object has the worst similarity ratings. To investigate further, participants were encouraged to comment on their ratings, particularly why they gave lower or higher ratings on some trials. Most of the participants who gave higher ratings mentioned that they did not perceive any unexpected haptic artifacts, noise, or instability. Additionally, they also mentioned that they were able to discriminate between the virtual feedback for different objects and their respective velocities. In contrast, several participants stated that when the real object falls on helmets, rather than obtaining feedback on one single contact point, they got feedback on a wide range of the helmet, which also made them feel uncomfortable. This effect was due to unconstrained interactions or collisions, whereas the impact feedback for the virtual object was rendered at a particular point.

From all the experimental results, it is observed that the data-driven approach provides higher realism than the physics-based approach. In particular, the data-driven approach demonstrates a very close recognition rate for plastic and wood type object matching compared to the real feedback matching study (see Fig. 14b). Similarly, the overall similarity scores of the data-driven approach were also relatively higher than the physics-based approach. This can be due to

the following reason. The data-driven approach renders a residual vibration signal just after the strong impulse, while the physics-based stimulus only contains the impulse at the moment of contact. We speculate from this that the residual vibration plays quite an important role in increasing the realism of impact rendering. This needs further investigation.

5.3 Performance of max–min extraction approach

This experiment is designed to examine the performance of the new max–min extraction approach (see Sect. 4.3), which reduces the dimensionality of 3D acceleration data, is compared with the state-of-the-art DFT321 approach. The DFT321 approach introduced in Park and Kuchenbecker (2019) takes 3-axis acceleration signals as input and synthesizes 1D impact signals $\hat{a}(t)$ that has the same spectral energy as the original signals. The process is accomplished by employing the below equations,

$$\hat{a}(t) = \hat{\mathcal{F}}[\tilde{A}(s)]\exp^{-j\theta(s)} \tag{12}$$

$$|\tilde{A}(s)| = \sqrt{|\tilde{A}_x(s)|^2 + |\tilde{A}_y(s)|^2 + |\tilde{A}_z(s)|^2} \tag{13}$$

$$\theta(s) = \tan^{-1}\left(\frac{\text{Im}(\tilde{A}_x(s) + \tilde{A}_y(s) + \tilde{A}_z(s))}{\text{Re}(\tilde{A}_x(s) + \tilde{A}_y(s) + \tilde{A}_z(s))}\right) \tag{14}$$

where $\tilde{A}_x(s)$, $\tilde{A}_y(s)$, and $\tilde{A}_z(s)$ are the three-axis acceleration vector components, $|\tilde{A}(s)|$ denotes the frequency-domain magnitude of DFT321 signal. Phase angles are represented by $\theta(s)$, which is determined by calculating the inverse tangent of the sum of the imaginary parts divided by the sum of the real parts of the original signals. Inverse Fourier transform function is represented by $\hat{\mathcal{F}}$. In Park and Kuchenbecker (2019), several approaches have been proposed for reducing 3D vibrations to 1D vibrations, and the DFT321 showed the best performance. Therefore, the present experiment uses the DFT321 approach as a benchmark for comparison.

Objective Evaluation Following the existing approaches (Landin et al. 2010; Park and Kuchenbecker 2019), we evaluate the numerical performance of DFT321 and Max–min extraction approach by employing the spectral match and

temporal match metrics. The spectral and temporal match metrics indicate the similarity between the collected 3D acceleration signals and the reduced 1D impact acceleration profile, which are explicitly defined in Landin et al. (2010); Park and Kuchenbecker (2019). Table 4 presents the object-wise comparison result between the DFT321 and max–min extraction approach. From this experiment, we can observe that in most cases max–min extraction approach performs best and preserves the original information in the signal.

Subjective Experimental Design This experiment had the same participants, experimental conditions, and procedure as the user study 1 with the data-driven method. The participants were provided with impact feedback using the proposed max–min extraction approach and DFT321 based approach. The feedback was rendered for four different virtual objects, listed in Table 1. Each participant carried out a total of 56 trials (4 objects \times 7 different velocities \times 2 actuators) for each approach. Afterward, they rated the feedback in terms of realism questionnaire. The rating was given in the form of 0–100 continuous scale.

Results Figure 16a shows the comparison results between the max–min extraction and DFT321 approach. Overall, the max–min extraction approach with vibro-transducer outperforms the other settings. Furthermore, the vibro-transducer with the max–min extraction approach had a statistically significant ($p < 0.01$) effect over the haptuator with both approaches. Moreover, we also compare the efficiency. Figure 16b shows the processing time for both the approaches. The processing time was measured during the conversion of each 3D acceleration signals to a 1D impact signal. The median processing time for the DFT321 approach was approximately thrice than the max–min extraction approach, showing that the max–min extraction approach is even faster.

6 Conclusion

The main aim of the proposed system is to render realistic impact feedback of falling objects during an earthquake in a virtual reality environment that can be used for safety-training simulation exercises. In this regard, we proposed a deep encoder–decoder based data-driven approach for realistic impact feedback rendering as well as the max–min extraction method for converting the captured 3D acceleration data

Table 4 Object-wise spectral match and temporal match comparison

Objects	DFT321		Max–min extraction	
	Spectral match	Temporal match	Spectral match	Temporal match
Plastic frame	0.97	0.80	0.97	0.78
Concrete brick	0.90	0.64	0.93	0.72
Piece of wood beam	0.93	0.84	0.97	0.88
Steel plate	0.88	0.72	0.90	0.80

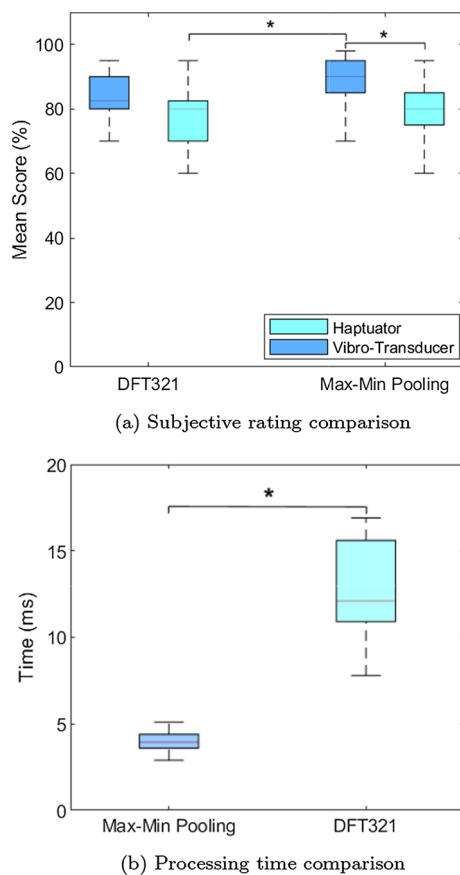


Fig. 16 Comparison between DFT321 and max–min extraction approach. Asterisks shows statistically significant difference with $p < 0.05$

into a 1D acceleration profile. The performance of the proposed approach was tested with the help of subjective evaluation and perceptual study. The results showed that the data-driven-based rendering outperformed baseline physics-based rendering, while participants preferred the vibro-transducer among the actuators. The proposed system still has room for further improvements. Currently, the system provides impact feedback on a single point (area); a design can be made to provide feedback on multiple locations of the body with multiple actuators. Additionally, in the future, for sports and workforce training, we will analyze rotational acceleration as well as linear acceleration in the form of impact feedback. To the best of our knowledge, our proposed framework could easily be extended to other impact feedback simulations, which may include impact of sports-related actions (e.g., kicking a football, hitting a ball with a bat), impact feedback during surgical training, the sensation of road bumps, and tapping on surfaces.

Acknowledgements This research was funded by the Preventive Safety Service Technology Development Program funded by the Korean Ministry of Interior and Safety under Grant 2019-MOIS34-001.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abdulali A, Jeon S (2016) Data-driven modeling of anisotropic haptic textures: data-segmentation and interpolation. In: Proceedings of the Eurohaptics. Springer
- Abdulali A, Rakhmatov R, Ogay T, Jeon S (2018) Data-driven modeling and rendering of force responses from elastic tool deformation. *Sensors* 18(1):237
- Abdulali A, Atadjanov IR, Jeon S (2020) Visually guided acquisition of contact dynamics and case study in data-driven haptic texture modeling. *IEEE Trans Haptics* 13(3):611–627
- Bortone I, Barsotti M, Leonardis D, Crecchi A, Tozzini A, Bonfiglio L, Frisoli A (2020) Immersive virtual environments and wearable haptic devices in rehabilitation of children with neuromotor impairments: a single-blind randomized controlled crossover pilot study. *J Neuroeng Rehabil* 17(1):1–14
- Chan LS, Choi, KS (2009) Integrating physx andopenhaptics: efficient force feedback generation using physics engine and haptic devices. In: Proceedings of the joint conferences on pervasive computing (JCPC), pp 853–858
- Culbertson H, Unwin J, Kuchenbecker KJ (2014) Modeling and rendering realistic textures from unconstrained tool-surface interactions. *IEEE Trans Haptics* 7(3):381–393
- Development of the number of earthquakes (m5+) worldwide from 2000 to 2019. <https://www.statista.com/statistics/263105/development-of-the-number-of-earthquakes-worldwide-since-2000/>. Accessed 7 July 2020
- Gallo D, Shreepriya S, Willamowski J (2020) Runahead: exploring head scanning based navigation for runners. In: Proceedings of the 2020 CHI conference on human factors in computing systems. ACM, pp 1–13
- Gilliland K, Schlegel RE (1994) Tactile stimulation of the human head for information display. *J Hum Factors Ergon Soc* 36(4):700–717
- Girbés V, Armesto L, Dols J, Tornero J (2016) Haptic feedback to assist bus drivers for pedestrian safety at low speed. *IEEE Trans Haptics* 9:345–357
- Gong V, Liu Y, Jiao Y, Wang B, Zhou J, Yu H (2015) A novel earthquake education system based on virtual reality. *IEICE Trans Inf Syst* E98.D:2242–2249
- Gunther S, Schon D, Muller F, Muhlhauser M, Schmitz M (2020) Pneumovolley: pressure-based haptic feedback on the head through pneumatic actuation. In: Proceedings of the CHI conference on human factors in computing systems. ACM, pp 1–10
- Handa T, Azuma M, Shimizu T, Kondo S, Fujiwara M, Makino Y (2019) Ball-type haptic interface to present impact points with vibrations for televised ball-based sporting event. In: Proceedings of the IEEE world haptics conference (WHC)
- Hassan W, Abdulali A, Jeon S (2020) Authoring new haptic textures based on interpolation of real textures in affective space. *IEEE Trans Ind Electron* 67(1):667–676

- Hosseini A, Richthammer F, Lienkamp M (2016) Predictive haptic feedback for safe lateral control of teleoperated road vehicles in urban areas. In: Proceedings of the vehicular technology conference
- Hwang JD, Williams MD, Niemeyer G (2004) Toward event-based haptics: rendering contact using open-loop force pulses. In: Proceedings of the 12th international symposium on haptic interfaces for virtual environment and teleoperator systems, pp 24–31
- Joolee JB, Jeon S (2022) Data-driven haptic texture modeling and rendering based on deep spatio-temporal networks. *IEEE Trans Haptics* 15(1):62–67
- Kaul OB, Rohs M (2017) Haptichead: a spherical vibrotactile grid around the head for 3d guidance in virtual and augmented reality. In: Proceedings of the 2017 CHI conference on human factors in computing systems. ACM, pp 3729–3740
- Kim S, Son B, Lee Y, Choi H, Lee W, Park J (2018) A two-DOF impact actuator for haptic interaction. In: Proceedings of the international AsiaHaptics conference, pp 173–177
- Kuchenbecker KJ, Fiene J, Niemeyer G (2006) Improving contact realism through event-based haptic feedback. *IEEE Trans Vis Comput Gr* 12(2):219–230
- Landin N, Romano JM, McMahan W, Kuchenbecker KJ (2010) Dimensional reduction of high-frequency accelerations for haptic rendering. In: Proceedings of the haptics: generating and perceiving tangible sensations (EuroHaptics), pp 79–86
- Li C, Liang W, Quigley C, Zhao Y, Yu L-F (2017) Earthquake safety training through virtual drills. *IEEE Trans Vis Comput Gr* 23:1275–1284
- Liang H, Liang F, Wu F, Wang C, Chang J (2018) Development of a VR prototype for enhancing earthquake evacuee safety. In: Proceedings of the 16th ACM SIGGRAPH international conference on virtual-reality continuum and its applications in industry. ACM, pp 1–8
- Lopes P, Ion A, Baudisch P (2015) Impacto: simulating physical impact by combining tactile stimulation with electrical muscle stimulation. In: Proceedings of the 28th annual ACM symposium on user interface software & technology. ACM, pp 11–19
- Lovreglio R, Gonzalez V, Feng Z, Amor R, Spearpoint M, Thomas J, Trotter M, Sacks R (2018) Prototyping virtual reality serious games for building earthquake preparedness: the Auckland city hospital case study. *Adv Eng Inform* 38:670–682
- Okada T, Okamoto S, Yamada Y (2021) Passive haptics: greater impact presented by pulsive damping brake of DC motor and physical indices for perceived impact. *Virtual Real* 25:233–245
- Osgouei RH, Kim JR, Choi S (2020) Data-driven texture modeling and rendering on electrovibration display. *IEEE Trans Haptics* 13(2):298–311
- Park G, Choi S (2017) A physics-based vibrotactile feedback library for collision events. *IEEE Trans Haptics* 10(3):325–337
- Park G, Kuchenbecker KJ (2019) Objective and subjective assessment of algorithms for reducing three-axis vibrations to one-axis vibrations. In: Proceedings of the IEEE world haptics conference (WHC). IEEE, pp 467–472
- Park C, Park S, Oh S, Choi S (2019) Realistic haptic rendering of collision effects using multimodal vibrotactile and impact feedback. In: Proceedings of the IEEE world haptics conference (WHC). IEEE
- Poorten EV, Yokokohji Y (2006) Rendering a rigid virtual world through an impulsive haptic interface. In: Proceedings of the international conference on intelligent robots and systems
- Pyo D, Yang T-H, Ryu S, Kwon D-S (2015) Novel linear impact-resonant actuator for mobile applications. *Sensors Actuators A Phys* 233:460–471
- Ramirez M, Kubicek K, Peek-Asa C, Wong M (2009) Accountability and assessment of emergency drill performance at schools. *Family Community Health* 32(2):105
- Romano JM, Yoshioka T, Kuchenbecker KJ (2010) Automatic filter design for synthesis of haptic textures from recorded acceleration data. In: Proceedings of the IEEE international conference on robotics and automation, pp 1815–1821
- Sagheer A, Kotb M (2019) Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems. *Sci Rep* 9(1):19038
- Seo SW, Kwon S, Hassan W, Talhan A, Jeon S (2019) Interactive virtual reality fire extinguisher with haptic feedback. In: Proceedings of the ACM symposium on virtual reality software and technology. ACM, pp 1–2
- Shin S, Choi S (2020) Hybrid framework for haptic texture modeling and rendering. *IEEE Access* 8:149825–149840
- Srivastava N, Mansimov E, Salakhutdinov R (2015) Unsupervised learning of video representations using LSTMs. In: Proceedings of the 32nd international conference on international conference on machine learning, pp 843–852
- Tsai HR, Chen BY (2019) Elastimpact: 2.5d multilevel instant impact using elasticity on head-mounted displays. In: Proceedings of the 32nd annual ACM symposium on user interface software and technology. ACM, pp 429–437
- Xu Z, Zhang H, Wei W, Yang Z (2019) Virtual scene construction for seismic damage of building ceilings and furniture. *Appl Sci* 9(17):3465
- Yim S, Jeon S, Choi S (2016) Data-driven haptic modeling and rendering of viscoelastic and frictional responses of deformable objects. *IEEE Trans Haptics* 9(4):548–559

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.