Distributed Haptic Interactions with Physically Based 3D Deformable Models over Lossy Networks

Ziying Tang, Yin Yang, Xiaohu Guo, *Member*, *IEEE*, and Balakrishnan Prabhakaran

Abstract—Researchers have faced great challenges when simulating complicated 3D volumetric deformable models in haptics-enabled collaborative/cooperative virtual environments (HCVEs) due to the expensive simulation cost, heavy communication load, and unstable network conditions. When general network services are applied to HCVEs, network problems such as packet loss, delay, and jitter can cause severe visual distortion, haptic instability, and system inconsistency. In this paper, we propose a novel approach to support haptic interactions with physically based 3D deformable models in a distributed virtual environment. Our objective is to achieve real-time sharing of deformable and force simulations over general networks. Combining linear modal analysis and corotational methods, we can effectively simulate physical behaviors of 3D objects, even for large rotational deformations. We analyze different factors that influence HCVEs' performance and focus on exploring solutions for streaming over lossy networks. In our system, 3D deformation can be described by a fairly small amount of data (several KB) using accelerations in the spectral domain, so that we can achieve low communication load and effective streaming. We develop a loss compensation and prediction algorithm to correct the errors/distortions caused by network problem, and a force prediction method to simulate force at users' side to ensure the haptic stability, and the visual and haptic consistency. Our system works well under both the client-server and the peer-to-peer distribution structures, and can be easily extended to other topologies. In addition to theoretical analysis, we have tested the proposed system and algorithms under various network conditions. The experimental results are remarkably good, confirming the effectiveness, robustness, and validity of our approach.

Index Terms—Collaborative virtual environments, 3D deformation, haptics, real-time simulation, user interaction

1 Introduction

Haptics-based user interaction has attracted increasing attention recently. Compared to traditional modes, haptics-based interaction provides a more intuitive and immersive approach for users to feel and manipulate virtual objects (e.g., feeling friction force or experiencing the texture of an object's surface [1], [2]), which a mouse-based interaction cannot provide. The kinesthetic feedback through haptic device makes application more interesting, and also significantly improves task performance [3], [4]. In addition, with the development of networks, studies of haptic interaction have been extended to distributed environments as well. Haptics-enabled collaborative/cooperative virtual environments (HCVEs) allow remote users to complete a task and share visual and haptic updates with each other collaboratively/cooperatively. This not only provides great flexibility for remote collaborations, but it also increases immersive experiences in distributed settings. As pointed out by Basdogan et al. [5], employing haptic feedback to support collaboration can considerably enhance copresence.

Although researches on HCVEs have been active in the past decade, there are still remaining challenges related to the 3D models used in interactions. We consider two primary types of 3D models: 1) *rigid/static* 3D objects, whose shapes remain the same regardless of the external force, and 2) *deformable* objects, whose shapes may change corresponding to the applied force. While it is relatively easy to interact with rigid/static 3D objects in HCVEs because only 6 degrees of freedom (DOFs) are needed to represent their motions, it is much more computationally expensive to interact with deformable objects, especially when involving physically based volumetric models, because the motion of a deformable object must be described by a large number of DOFs governed by physical principles related to strain tensor. Therefore, real-time deformable simulation in a distributed environment is challenging, and the related consistency problems have not been very well studied. Further, if a collaborative system is running over general networks, additional network-related issues are non-neglectable, as packet loss,[1] delay,[2] and jitter[3] are still very

- *Z. Tang is with the Department of Computer and Information Sciences, Towson University, 8000 York Road, Towson, MD 21252. E-mail: ztang@towson.edu.*
- *Y. Yang is with the Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, NM 87131. E-mail: yangy@unm.edu.*
- *X. Guo and B. Prabhakaran are with the Computer Science Department, University of Texas at Dallas, 800 West Campbell Road, Dallas, TX 75083. E-mail: xguo@utdallas.edu, praba@utdallas.edu.*

Manuscript received 27 July 2012; revised 4 July 2013; accepted 12 Aug. 2013; published online 18 Sept. 2013.
Recommended for acceptance by M. Otaduy.
For information on obtaining reprints of this article, please send e-mail to: toh@computer.org, and reference IEEECS Log Number TH-2012-07-0055.
Digital Object Identifier no. 10.1109/ToH.2013.47.

1. Packet loss occurs when packets of data fail to reach their destination, when they are transmitted over unstable networks. The reason includes a number of factors such as channel congestion, signal degradation, and so on.
2. Packet delay refers to the transmission delay over networks. It is the amount of time required to transfer data to its reception.
3. Jitter implies the packet delay variation. It measures the variability over time of the latency. A network with constant latency does not have jitter.

1939-1412/13/$31.00 © 2013 IEEE Published by the IEEE CS, RAS, & CES
Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

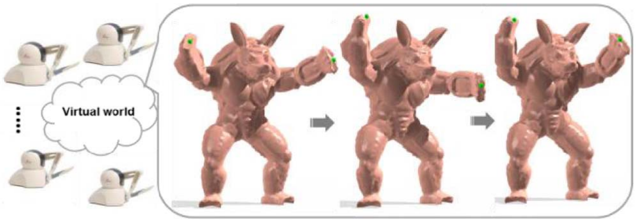


Fig. 1. An example of the proposed system.

common over a nondedicated channel, even though the network bandwidth has been increased significantly. Those network problems can cause very serious visual distortion, especially for time-dependent deformable simulations. Furthermore, when involving haptics to provide a highly interactive response for users, it is compulsory that the force computation is completed efficiently, as an update rate of at least 1 KHz is necessary to maintain a stable force feedback [6]. However, ensuring the high update rate in distributed setting is very difficult, as the unstable network conditions cannot guarantee high-speed data transmission without data loss. Thus, haptic instability and system inconsistency posit another problem we have to consider.

Previous researches have proposed different methods to share 3D simulation in real time [7], [8], [9]. However, most of them focus on rigid models or simple surface deformable models (e.g., mass-spring-based), and sharing haptic interaction with complicated physically based solid models in HCVEs has not been well addressed. We believe involving complex volumetric data in an HCVE can expand the study scope and facilitate development of corresponding applications such as virtual education and assembly. Studies have also been made on different architectures of HCVEs and focused on exploring solutions for network delay [7], [8], [10], [11]. But there are not many researches on other network-related issues, especially packet loss, which is a very important problem to address.

1.1 Proposed Approach

In this paper, we propose a novel approach to support haptic interactions with physically based 3D deformable models in a distributed virtual environment. The objective is to achieve real-time sharing of deformation and force simulations over general networks where packet loss, delay, and jitter can happen. Fig. 1 illustrates the proposed framework: multiple users at various locations cooperatively manipulate the same soft object through their haptic devices and receive visual and force feedback simultaneously.

To achieve real-time computational performance, we choose the linear modal analysis [12], to provide an efficient and natural way to model physical behaviors of 3D objects. In addition, we follow a corotational method, called Modal Warping [13], to overcome the limitation of the linear modal analysis and handle large rotational deformations with linear strain tensor. Futhermore, we analyze different factors that influence an HCVE's performance, and focus on exploring solutions for streaming over lossy networks. In our system, 3D deformation is described by a fairly small amount of data (several KB) using accelerations in the modal subspace, which considerably reduces communication load and improves streaming efficiency. To correct the errors/distortions caused by data loss, delay, and jitter, we propose a loss compensation and prediction algorithm that ensures smooth and physically correct simulation. Moreover, we develop a force prediction method for completing haptic calculations at users' side to fulfill the requirements of synchronization and high-speed haptic simulation. The proposed system is discussed over different distribution strategies and tested under various network conditions. The experimental results prove the robustness and efficiency of our system and algorithms.

An earlier version of this paper appears in [14], where we have explained the deformation and force simulation. In this version, we extend it by studying different system impact factors and focusing on solving the network-related problems. Unlike [14], we describe the deformation updates using acceleration. Besides, system consistency and the synchronization issue between visual and haptics simulation are also discussed in this paper.

1.2 Contributions

This work makes several important contributions to the literature of HCVE researches. First, we extend the study of collaborative haptic interactions into the physically based 3D deformation. Second, we develop a method to compute interactive forces and deformations in real time so that distributed interactions in an HCVE can be accomplished efficiently. Third, we propose a new compensation and prediction algorithm to tolerate errors and inconsistencies caused by packet loss, delay, and jitter when streaming data over unstable channels. This allows the system to work well in general network settings. Last, in addition to a theoretic analysis of performance impact factors, we conduct experiments to evaluate the performance of the proposed system and algorithms.

Our current system supports linear elasticity (with corotational Modal Warping) in the simulation algorithm. It can be extended to handle nonlinear deformation with Green tensor, but cannot support viscoelastic deformation used in some medical simulations. Our delay/loss compensation method is based on the linearity property of numerical time integration methods, which can be easily extended to other deformation algorithms.

The remainder of this paper is organized as follows: After reviewing related works in Section 2, we discuss the performance impact factors in Section 3. The force and deformation computation is explained in Section 4. Section 5 describes the loss compensation and prediction algorithm and two system architectures. Section 6 presents experimental results and related discussions. Section 7 concludes this paper with limitations and future works.

2 Related Works

Simulation of deformable models has been an important research topic in computer graphics since early 1980s. The realism of simulation and the computational performance are two major considerations in deformable model simulation. Pioneering work in physically based deformable simulation is attributed to Terzopoulos and his coworkers [15], [16]. Then, a large number of mesh-based methods for both offline and interactive simulation have been proposed

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

based on either the boundary element method [17] or the finite element method [18]. Among different methods, linear elasticity models have been popular because they are stable and computationally efficient. However, they cannot handle large rotational deformations, which require the use of nonlinear Green tensor. Some approaches [16], [19] treat deformations as a combination of global rigid motion and local linear elastic deformation, but precise simulation of large rotations still cannot be achieved. Corotational methods, which keep tracks of rotational part locally to compute linear elastic force, provide a good solution for large deformations and have been widely used [20], [21], [22].

From another perspective, some previous works propose to improve efficiency through dimensional reduction, called modal analysis. Pentland and Williams [12] introduce a modal analysis framework to decompose the deformation space into a set of vibration modes. They have pointed out that the modes associated with higher resonance have less effect on the shape of the object. Based on it, Hauser et al. [23] further extend this framework by integrating the manipulation, collision, and other constraints. Basdogan [24] explains how it can be applied to achieve real-time medical simulation and compares it with the spectral Lanczos decomposition method. Modal analysis has also been used to synthesize geometrically complex deformation using graphics hardware [25]. Similarly, Raghuvanshi et al. [26] also utilize graphics hardware to further improve simulation performance. Based on modal analysis, Choi and Ko [13] propose a corotational method called modal warping to support large deformations by utilizing a modal rotation matrix to calculate the node-wise rotation of the whole deformable object. They have shown that the computation based on linear strain tensor can handle large rotational deformation. This technique has been extended into meshless and thin-shell simulation problems [27], [28], and hybrid solid simulation [29]. Similar idea has also been employed for the computation and streaming of deformable surfaces using Manifold Harmonics [30], [31].

Haptic rendering for soft objects has long been an active research area. Single point haptic interaction with deformable object has been studied from 1990s [32], [33]. Then volumetric model is introduced to haptic rendering in some research works, focusing on how to achieve real-time rendering speed by using some preprocessing techniques [34], [35]. Barbič and James [36] propose a multi-interaction haptic rendering method that achieves real-time speed by using a multiresolution point-shell construction. Mafi et al. [37] suggest a hardware-based parallel computing approach for real-time haptic interaction with deformable bodies. FEM-based soft tissue simulation has been applied to improve haptic simulation [32]. To improve simulation speed, history-independent deformation is widely accepted and has been applied to different applications. For example, Qin et al. [38], and Ullrich and Kuhlen [39] have separately proposed haptics-based virtual surgery simulation using the history-independent deformation. Nonlinear visio-haptic interaction with soft model has also been studied [40]. It is true that haptic rendering for deformable models is a well-developed area, and various approaches have been proposed from different perspectives. However, when considering haptic rendering in a distributed environment, there are still remaining challenges.

The prior studies on haptic streaming and collaboration mainly focus on two research topics: synchronization of virtual scenes, and reliability and effectiveness of haptic feedback. Different topologies, including client-server (C-S) [10], [38], [41], peer-to-peer (P2P) [7], [42], and hybrid type [8], have been explored to build collaborative haptic applications with various objectives. For example, Marsh et al. [8] analyze various network architectures and suggest a hybrid topology to support rich collaborative behaviors. Iglesias et al. [7] study scene synchronization problems in a P2P structure. On the other hand, some approaches suggest reducing the network latency to improve the haptic streaming performance. For example, Al Osman et al. [43] develop an application layer protocol named ALPHAN that uses XML-based descriptions and supports multi-buffering for haptic data communication. Most of the previous literatures, such as those mentioned above, focus on haptic interaction with static/rigid objects. There are some researches on remote haptic interaction with deformable models [9], [38]. However, they either use mass-spring deformable objects whose behavior is limited or do not consider general network conditions where synchronization and reliability are crucial.

3 Factors Affecting Distributed Interactions

A distributed virtual system that supports rich behaviors such as the simulation of physically based deformation and haptic interactions can be very complex. Its performance is determined by different factors: interaction modes, distribution strategies, network conditions, transmission protocols, simulation latency, and so on.

In a groupware application, remote users' interaction mode is very essential when analyzing the demands imposed on the system. Buttolo et al. [10] classify user interaction into two types, namely *collaborative* and *cooperative* modes. The first one means participants take turn to perform interaction, for example, collaborative assembling. In this mode, there is always only one active user at a time period, so it is easier to ensure the system consistency and the simulation is simpler. The cooperative mode, on the other hand, implies that multiple users concurrently interact with the same object, such as cooperative grasping or lifting, so it is much complex and has higher requirements on communication latency and stability. Our HCVE is able to successfully support both collaborative and cooperative interactions. For the ease of description, we simply use the term *collaboration* in later sections.

The system architecture of an HCVE is another performance impact factor. As pointed out by Iglesias et al. [7], there is no architecture that is optimal for all networked environments and the system structure and its performance is decided by the application's specific goals. Commonly used strategies include client-server (C-S), peer-to-peer (P2P), and hybrid types. The C-S architecture is robust against the effect of network jitter. A semantically consistent

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

state can be easily achieved by using a centralized server, so the C-S architecture has been regarded as the most appropriate one to support cooperative tasks [10]. The P2P structure, on the other hand, successfully minimizes latency, but suffers from network jitter, because the maximum latency from the slowest network may be propagated to others to achieve full synchronization [8].

Common network problems include packet loss, delay, and jitter. Their influences on a distributed system vary, depending on the virtual content, network structure, and the goal of application. For a system involving only visual display like video, an update rate of 20-30 Hz is typically sufficient to yield a smooth and stable result, implying that its tolerance of network problems is high. Haptic rendering in an HCVE, however, requires an update rate of 1 KHz so that the network delay could cause serious instability. As indicated in [8], a latency of 25-30 ms can lead to unusual kinesthetic feeling. Applications such as virtual surgery have lower tolerance on packet loss as compared to online games. A packet loss may only bring a short visual distortion to online games, but can cause totally wrong feedback to virtual surgery. Network problems are tightly related with bandwidths. In a highly congested network with small available bandwidth, packet drop occurs frequently and packet delivery experiences longer queuing time, leading to longer delay. It also leads to high degree of jitter as different routers may be chosen to avoid the congested link. The transmission load is directly determined by the bandwidth, so streaming large-sized data is more likely to experience network problems.

The communication protocol is another issue influencing the system performance. Reliable protocols like TCP ensure packets are delivered in the correct order. Thus, they can be applied to avoid problems caused by transmission loss and disorder. However, using TCP introduces additional communication latency due to retransmission and the subsequent packets in the stream have to be queued until the lost ones arrive. In contrast, unreliable protocols such as UDP do not detect delivery order and packet loss, but they can provide faster streaming. Therefore, we prefer to utilize UDP channel for our real-time system, and handle data loss with separate modules.

Simulation time is another aspect that affects HCVEs' performance. It is not a major problem when rigid or simple deformable objects are involved. However, simulating physically based deformation and haptic interactions can be very time-consuming. It poses another challenge to the proposed HCVE: How to reduce simulation time to achieve real-time collaboration?

4 Deformation and Force Simulation

A linear modal analysis technique is applied in our system to simulate physically based deformation and force. Below we explain these simulations in detail.

4.1 Physically Based Simulation

The motion of physically based 3D deformable models is time-dependent and governed by physical principles described by a constitutive law relating forces to strains. To integrate the corresponding ordinary differential equations (ODE) over time, we subdivide the time axis into small pieces, for example, 1/40 second, called *time-steps*. At every time-step, in addition to the position, velocity, and other physical properties of every vertex, we compute the force feedback as a response to users' interaction.

The deformation is triggered by users' manipulation through some interaction points on the 3D object, called *constraint points*. By dragging those points to target positions, users can trigger deformation and change the shape and motion of a virtual object. Based on this, we compute how much force is required to maintain the corresponding deformation, which is called *constraint force* and is treated as the haptic feedback. Our HCVE supports multiple users to collaboratively/cooperatively interact with one common model. Each user controls one constraint point and receives the force feedback at that point; meanwhile, all users are able to view and feel the change of shape and motion under others' interactions.

4.2 Spectral Force and Deformation

To achieve real-time performance in an HVCE, it requires low computational cost and small streaming load. For this purpose, we follow a linear modal analysis method [12], [29], which utilizes a very small number of modal/spectral bases to describe the motion, to reduce the DOFs and increase the performance considerably. In addition, to support large rotational deformation with linear strain tensor, we follow the corotational algorithms [20], [21], [22] and adopt the Modal Warping technique [13] in the system. The idea behind the Modal Warping method is based on the infinitesimal rotation tensor. It keeps track of the local rotations during deformation, and warps precomputed modal basis with local rotation at each time-step. Since force and deformation are causally related, we propose to compute force using modal/spectral bases as well. Thus, simulations of both force and deformation are performed in the spectral domain in our HCVE. Below, we briefly explain the simulation with the summary of main notations listed in Table 1. The detailed derivation can be found in the online supplementary material at the Computer Society Digital Library http://doi.ieeecomputersociety.org/10.1109/TOH.2013.47.

The governing equation, i.e., the Euler-Lagrange equation, of a 3D deformable body discretized using the finite-element method (FEM) is

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \tag{1}$$

where $\mathbf{u}$ is a $3n$ time-dependent vector representing the displacement of the whole deformable body (n vertices) from the original position; matrices $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are independent of time under the linear elasticity assumption, and determined by the object's physical properties.

Following Modal Warping, we integrate the local rotations occurring at nodal points to get the rotation matrix $\mathbf{R}$. Then, using the $3n \times m$ modal displacement matrix $\mathbf{\Phi}$, we rewrite (1) as

$$\mathbf{M}_q\ddot{\mathbf{q}} + \mathbf{C}_q\dot{\mathbf{q}} + \mathbf{K}_q\mathbf{q} = \mathbf{f}_q, \tag{2}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ denote the *spectral displacement*, *velocity*, and *acceleration*, respectively, and $\mathbf{f}_q = \mathbf{\Phi}^T(\mathbf{R}^T\mathbf{f})$ is the *spectral external force*. Note that the number of frequency

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Summary of Math Notations

Notation	Definition
$\mathbf{M}(\mathbf{M}_q)$	The spatial (spectral) mass matrix
$\mathbf{C}$ ($\mathbf{C}_q$)	The spatial (spectral) damping matrix
$\mathbf{K}$ ($\mathbf{K}_q$)	The spatial (spectral) stiffness matrix
$\mathbf{\Phi}$	The model displacement matrix
$\mathbf{R}$	The rotation matrix
$\mathbf{u}/\dot{\mathbf{u}}/\ddot{\mathbf{u}}$	The spatial displacement/velocity/acceleration
m	The frequency modes used for simulations.
h	The size of the time step
$\mathbf{q}^t/\dot{\mathbf{q}}^t/\ddot{\mathbf{q}}^t$	The spectral displacement/velocity/acceleration at time step t
β, γ	Constant scalar: $\beta = 1/4$ and $\gamma = 1/2$
$\mathbf{c}$	The position vector of constraint points
$\mathbf{f}$ ($\mathbf{f}_q$)	The spatial (spectral) constraint force
$\Delta\mathbf{q}^{j-i}$	The displacement error. Packet loss occurs at the time step i and j is a time step after i
$\Delta\dot{\mathbf{q}}^{j-i}$	The velocity error, similar as displacement error
$\hat{\ddot{\mathbf{q}}}^i$	The predicted acceleration at time step i

modes (denoted as m) used for calculation is significantly smaller than n. For the sake of numerical stability, an implicit integration algorithm [44] is used. Then, the unconstrained motion that describes the initial status of a 3D object is represented as

$$\mathbf{A}\ddot{\mathbf{q}} = \mathbf{b}, \tag{3}$$

where $\mathbf{A} = \mathbf{M}_q + \gamma h\mathbf{C}_q + \beta h^2\mathbf{K}_q$, $\mathbf{b} = \mathbf{f}_q^{t+1} - \mathbf{C}_q\tilde{\dot{\mathbf{q}}}^{t+1} - \mathbf{K}_q\tilde{\mathbf{q}}^{t+1}$, and $\tilde{\mathbf{q}}^{t+1}$ and $\tilde{\dot{\mathbf{q}}}^{t+1}$ are predictors of the spectral displacement and velocity, respectively.

Users' interactions are represented by the desired positions of constraint points in a vector $\mathbf{c}$, so the system under users' manipulation can be described as

$$\begin{pmatrix} \mathbf{A} & \tilde{\mathbf{J}}^T \\ \tilde{\mathbf{J}} & 0 \end{pmatrix}\begin{pmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \tilde{\mathbf{c}} \end{pmatrix}, \tag{4}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, $\tilde{\mathbf{J}} = \beta h^2\mathbf{J}$ and $\tilde{\mathbf{c}} = \mathbf{c} - \mathbf{J}\tilde{\mathbf{q}}$. Matrix $\mathbf{J}$ is the generalized constraint satisfying $\mathbf{J}\mathbf{q} = \mathbf{c}$. Solving (4) gives us the solution of the spectral acceleration, $\ddot{\mathbf{q}}$, which describes the object's movement in modal subspace. Note that the size of spectral acceleration is m, which is typically very small.

Based on (4), we get the force at the constraint nodes, and call it the *spectral constraint force*, $\mathbf{f}_q = \tilde{\mathbf{J}}^T\boldsymbol{\lambda}$. We can then define the spatial *constraint force*, $\mathbf{f}$, and treat it as the haptic feedback,

$$\mathbf{f} = \mathbf{R}\mathbf{\Phi}\mathbf{f}_q. \tag{5}$$

5 Loss/Delay Compensation and Prediction

As explained previously, the force and deformation simulations can be accomplished quickly in the modal subspace, based on which we build our HCVE and discuss how it works over lossy network conditions.

5.1 Collaborative System

Generally, our HCVE can support any distributed structure. Below, we explain how it works in two fundamental topologies, i.e., the client-server (C-S) and the peer-to-peer (P2P), and the data streaming in each topology.

Depending on the structure, two types of data may be streamed, namely *user interactions*, which are represented by the desired constraints ($\mathbf{c}$), and *deformation updates*, which refer to the spectral accelerations ($\ddot{\mathbf{q}}$).

We use the spectral acceleration instead of the spectral displacement ($\mathbf{q}$) to describe deformation update because of the following two reasons. First, although displacement is the information that users can directly perceive, the users' feeling of "physics" comes from the fact that the displacement is integrated from velocity and acceleration, i.e., it is second-order continuous w.r.t. time. Directly stream displacements will affect users' perception when the displacement is lost or delayed. Thus, representing the deformation with acceleration and integrating it to get displacement will guarantee the visual quality of physics-based deformation under packet loss scenarios. Under this framework, we propose a loss compensation algorithm for recovering accelerations and displacements (in Section 5.3) and simulating forces (in Section 5.4), under general packet loss, delay, and jitter situations. Second, representing the deformation using acceleration can potentially achieve higher compression, and reduce the communication load in the C-S structure. When the acceleration is constant over time, like objects falling under gravitation, the information only needs to be streamed for the first frame. In this paper, we focus on handling packet loss, delay, and jitter situations and do not exploit this second advantage.

In the C-S structure, as illustrated in Fig. 2a, there is one centralized server that computes deformation by solving (4) based on user interactions, and broadcasts the updates to all the clients. Clients send their interactions to the server, and when receiving deformation updates they renew their local model representations and compute force locally using (5) or (10). In the P2P structure, as shown in Fig. 2b, every peer serves as a server as well as a client. Each peer simulates deformation and force at its own site by solving (4) and (5), or (10), based on its own and other peers' interactions. User interactions are the only data streamed in this architecture. It looks redundant to require all peers to perform computation. However, the advantage of the P2P structure is obvious: it has less data streaming and does not have a central server which may become bottleneck of the whole HCVE system.

5.2 Packet Loss, Delay, and Jitter

We consider how the proposed HCVE works in a nondedicated network environment, where packet loss, delay, and jitter happen. These network-related problems can cause serious visualization distortions, haptic instability, and system inconsistency.

To achieve real-time streaming, we utilize UDP protocol, so a packet loss means the failure of data delivery with no recovery and causes incomplete streaming result on the

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

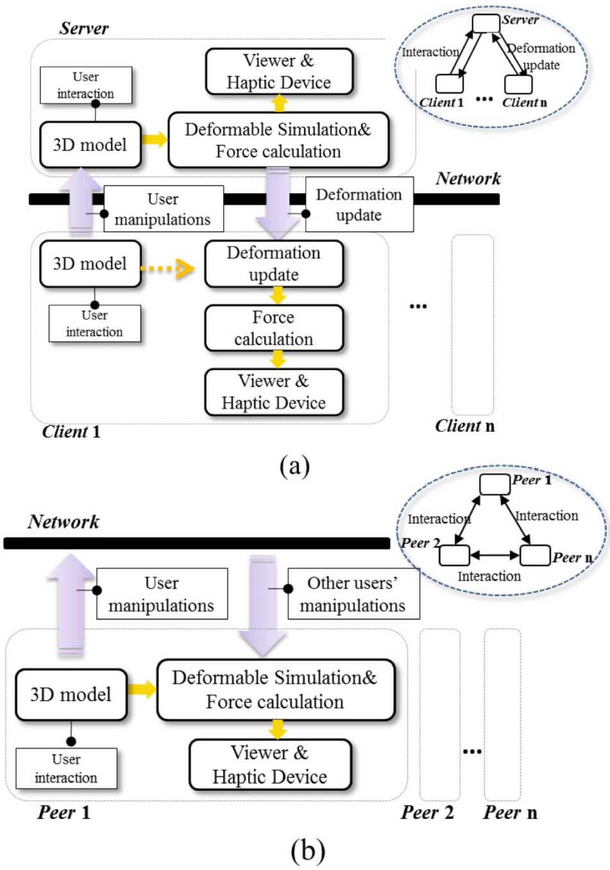


Fig. 2. Proposed collaborative system over (a) the Client-Server (C-S) and (b) the Peer-to-Peer (P2P) architectures.

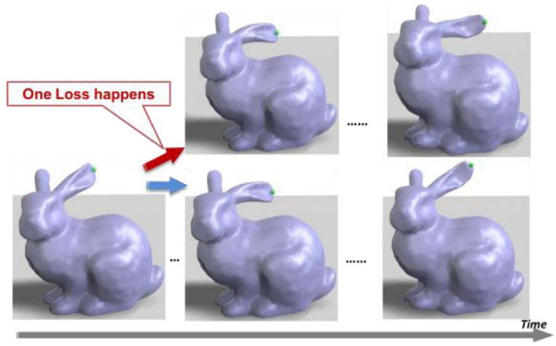


Fig. 3. One packet loss causes obvious deformation distortion.

receiver side. Thus, when a packet loss happens, a negative acknowledgement (NACK) message is employed to let the sender resend the lost data. By doing so, we convert packet loss into a delay problem. The main reasoning here is by using NACK we eventually try to get the original data, and then based on it we adjust simulations following a prediction. The jitter problem can be handled similarly. Without any loss of generality, we only discuss packet loss problem in later explanation.

When users' interaction is lost in the C-S architecture, the central server can still ensure system consistency, i.e., all clients have the same outputs as the server [8], [14]. Particularly, following a time-stamping method, interactions from different clients are maintained according to the generating orders, and then the deformation is computed sequentially by the central server. Note that the concurrent cooperative interactions from different users perform well in our HCVE, as each user interacts and receives haptic feedback at his/her own constraint point, assuming there is no conflict in selecting the same constraint point between different users.

As mentioned before, we stream deformation updates described by spectral acceleration in the C-S structure. If data loss happens, it causes visualization distortions and system inconsistency. As illustrated in Fig. 3, even one packet loss can lead to significantly different results. Thus, recovering lost data is crucial. Solutions like simple retransmission, however, cause not only additional delays, but also errors to the system. Because physical simulation is time-sensitive, delayed delivery does not help real-time simulation. Moreover, an acceleration loss at some time-step t affects velocities and displacements in the later time-steps, and these errors accumulate along time. How to correct simulation errors while maintaining a smooth and physically meaningful result is the key challenge. To address it, we propose a compensation and prediction algorithm described next. The key idea is instead of freezing the simulation procedure and waiting for the lost data to arrive, we predict it, and when the resent packet arrives, we correct errors using a compensation algorithm.

Packet loss issue in the P2P structure can be handled similarly using the proposed algorithm. In the P2P HCVE, user interaction is the only data streamed. Because user interactions determine the acceleration and deformation update, streaming user interaction in the P2P topology has the same result as directly streaming acceleration. In other words, the loss of user interaction in this structure has the same consequence as the loss of deformation update in the C-S architecture. Thus, it can also be solved following the proposed compensation algorithm.

5.3 Compensation Algorithm

As described previously, two predictors $\tilde{\mathbf{q}}^{t+1}$ and $\tilde{\dot{\mathbf{q}}}^{t+1}$ are used to compute unknown displacements and velocities. Based on them, we get the following two equations (the detailed derivation is included in Appendix A, which can be found in the online supplemental material):

$$\mathbf{q}^{t+1} = \mathbf{q}^{t} + h\dot{\mathbf{q}}^{t} + \frac{h^2}{2}\left(\frac{1}{2}\left(\ddot{\mathbf{q}}^{t} + \ddot{\mathbf{q}}^{t+1}\right)\right), \tag{6}$$

$$\dot{\mathbf{q}}^{t+1} = \dot{\mathbf{q}}^{t} + h\left(\frac{1}{2}\left(\ddot{\mathbf{q}}^{t} + \ddot{\mathbf{q}}^{t+1}\right)\right). \tag{7}$$

Equations (6) and (7) imply an interesting result: the displacement $\mathbf{q}$, the velocity $\dot{\mathbf{q}}$, and the acceleration $\ddot{\mathbf{q}}$ at two continuous time-steps are linearly related. This reveals an important relationship between acceleration and velocity change, as summarized below (the proofs of lemma and theorem are included in Appendix B, which can be found in the online supplemental material):

Lemma. *The acceleration at time-step t ($\ddot{\mathbf{q}}^{t}$) only contributes to the change of velocity at time-steps t and $t+1$.*

The lemma shows that the acceleration has no effect on velocities after two time-steps. Thus, the effect of an

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

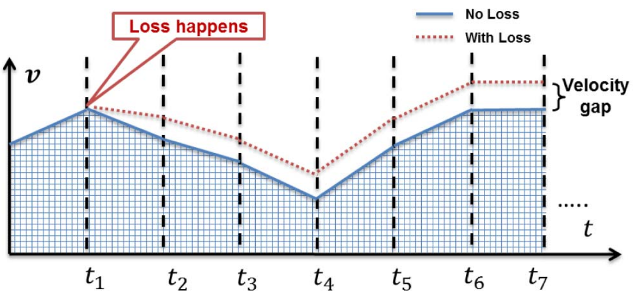


Fig. 4. Displacement and velocity errors along time. When data is lost at t_1, velocity changes along red dash line.

acceleration loss on velocity and displacement is limited. Based on the lemma, we examine the displacement error caused by the acceleration loss and call it *displacement distortion*, denoted by $\Delta\mathbf{q}$. The result is summarized in the following theorem.

Theorem. *When the acceleration at time-step t ($\ddot{\mathbf{q}}^{\mathrm{t}}$) is lost, the displacement distortion ($\Delta\mathbf{q}$) becomes linear with respect to time from time-step t+2.*

Fig. 4 illustrates the theorem using an example. Particularly, if the acceleration of time-step t_1 ($\ddot{\mathbf{q}}^{t=1}$) is lost, the velocity curve changes from the blue-solid line to the red-dashed line. After two time-steps, the solid and dashed lines are parallel to each other, even though the displacement distortion which is shown by the shadow part in the figure accumulates along time. Based on the lemma and theorem, we conclude that it is not necessary to record the acceleration history of the object to achieve resynchronization. Instead, the only information needed is the time-step when packet loss happens. Moreover, there is no need to roll back the simulation (and we should not do so), as the simulation error can be corrected and the system can be resynchronized using the algorithm below, although users may experience different results in the short period from data loss to compensation.

Supposing that the acceleration is lost at time-step i, based on (6) and (7), the distortion at any later time-step j can be computed as follows:

$$\Delta\dot{\mathbf{q}}^{\mathrm{j-i}} = \begin{cases} \frac{1}{2}h\ddot{\mathbf{q}}^{\mathrm{i}} & \text{if j} = \text{i} \\ h\ddot{\mathbf{q}}^{\mathrm{i}} & \text{if j} > i, \end{cases} \tag{8}$$

$$\Delta\mathbf{q}^{\mathrm{j-i}} = \begin{cases} \frac{1}{4}h^2\ddot{\mathbf{q}}^{\mathrm{i}} & \text{if j} = \text{i} \\ (\mathrm{j}-\mathrm{i})h^2\ddot{\mathbf{q}}^{\mathrm{i}} & \text{if j} > i, \end{cases} \tag{9}$$

where $\Delta\dot{\mathbf{q}}^{\mathrm{j-i}}$ and $\Delta\mathbf{q}^{\mathrm{j-i}}$ are the distortions of velocity and displacement, respectively. Note that if more than one packet is lost, the errors are the summation of all losses. Based on the previous analysis, we propose the loss compensation algorithm as follows:

1. If a packet loss is detected at time-step i, notify the sender to resend the data using a NACK message;
2. Record the lost time-step *i*;
3. Predict the lost data or set it as zero;
4. Compute deformations; and

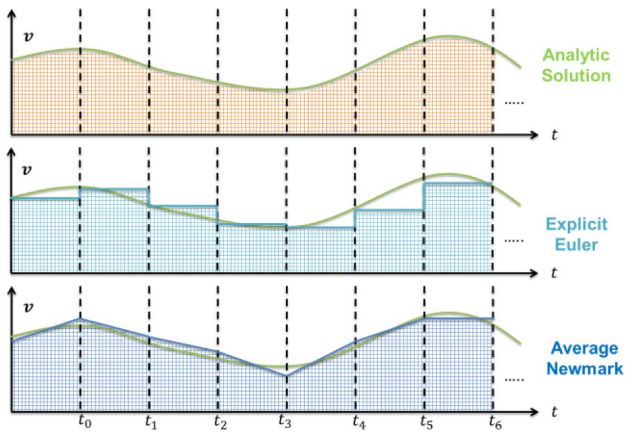


Fig. 5. Numerical integration over time assumes either velocity or acceleration to be constant within one time-step.

5. When the resent packet arrives at time-step j, compensate velocity and displacement differences in s steps using (8) and (9).

In step 3, we can either set the lost acceleration as zero, or predict it based on previous history using techniques such as linear extrapolation. When using prediction, $\ddot{\mathbf{q}}^{\mathrm{i}}$ will be changed to $\Delta\ddot{\mathbf{q}}^{\mathrm{i}}$ in (8) and (9), where $\Delta\ddot{\mathbf{q}}^{\mathrm{i}} = \ddot{\mathbf{q}}^{\mathrm{i}} - \widehat{\ddot{\mathbf{q}}}^{\mathrm{i}}$, and $\widehat{\ddot{\mathbf{q}}}^{\mathrm{i}}$ is the predicted acceleration at time-step i. In step 5, compensation could be completed in s (s>1) time-steps instead of one. Doing so allows us to gradually correct distortions to achieve a much smoother and natural simulation result.

It is worth noting that our compensation algorithm is based on the nature of linearity of numerical time integration rather than the linearity of deformation method. So it can be easily extended to other deformation methods (linear/nonlinear elasticity, etc.). No matter what deformation algorithm is applied, we can handle the network-related problems with a similar mechanism.

To get the shape of the object after deformation, we need to compute the displacement $\mathbf{u}$ for each DOF. Analytically the solution of $\mathbf{u}$ is $\mathbf{u}(t_0) = \int_0^{t_0} \dot{\mathbf{u}}dt$ and $\dot{\mathbf{u}}(t_0) = \int_0^{t_0} \ddot{\mathbf{u}}dt$, where according to Newton's second law, $\ddot{\mathbf{u}} = \frac{\mathbf{f}_{external} - \mathbf{f}_{damping} - \mathbf{f}_{internal}}{m}$, (note, m is mass here). However, the external force $\mathbf{f}_{external}$, damping force $\mathbf{f}_{damping}$, and the nonlinear elastic internal force $\mathbf{f}_{internal}$ are unavailable in the analytical form. Therefore, numerical method is needed to compute the unknown displacement for each DOF no matter what type of deformation algorithms are employed. Fig. 5 demonstrates the above discussion in a picture. The analytic solution of the displacement is the shadowed orange region in the top diagram. No matter what type of numerical time integration is adopted, to linearize the ODE, either first- or the second-order derivative of the displacement ($\dot{\mathbf{u}}$, $\ddot{\mathbf{u}}$ as in explicit Euler or average Newmark as shown in Fig. 5) is used. Therefore, a packet loss leads to a velocity gap as shown in Fig. 4. This velocity gap causes an accumulation of displacement deviation for the DOF, which is linear to the length of time between package-loss and package-rearrival. Such accumulated loss can always be computed similarly using the proposed compensation algorithm if data loss is detected.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

Thus, the proposed compensation method is related to the type of numerical time integration adopted in the simulation, and it can be extended to any deformation algorithm that linearizes the ODE within time-steps to approximate the target integration.

5.4 Force Simulation and Prediction

Packet loss can bring problems not only to deformation, but also to force simulation if both are streamed over the network [14], and the impact on haptic simulation is more critical. Unlike acceleration, force is a time-sensitive information and the influence of a packet loss on force update is only active at that time-step rather than accumulating over time. In other words, one force loss at time i will not affect any force simulation after time i. However, this short-term loss is very problematic: it brings sudden kinesthetic change which can cause instabilities to the whole system and impact users' ability to coordinate their actions. Moreover, retransmitting the lost force or waiting for the delayed force to arrive does not help ensure timely feedback. In contrary, waiting for delayed force update could cause misperception. For example, user may believe he/she makes some mistakes due to the sudden force change, even though he/she actually performs correct operations [8].

Another question needs to be considered here is how to keep the consistency between deformation and force, i.e., the correlation of visual and force feedback. A loss of acceleration data leads to continuous distortions on deformation, while a force loss only impacts haptic feedback at one time-step. Consequently, we may have mismatched visual and force feedback. For example, we may feel a strong force on one direction but the model doesn't move or moves wrongly. To address these problems, we suggest performing force simulation on the receivers' side instead of streaming it over the lossy networks. Because of the applied linear modal analysis simulation algorithm in the proposed HCVE, we are able to compute force based on deformation. So, no matter in which network topologies, the deformation update and/or user interactions are the only streamed data. When packet loss happens and the proposed compensation method is applied, we follow a receiver-based force simulation algorithm:

1. At time-step i, if a packet loss is detected, predict the deformation/acceleration using the loss compensation algorithm described in Section 5.3.
2. Based on the predicted acceleration, simulate the force using,

$$\mathbf{A}\ddot{\mathbf{q}} + \mathbf{f}_{\mathrm{q}} = \mathbf{b}, \tag{10}$$

where $\mathbf{f}_{\mathrm{q}}$ is spectral force, $\mathbf{A} = \mathbf{M}_{\mathrm{q}} + \gamma h \mathbf{C}_{\mathrm{q}} + \beta h^2 \mathbf{K}_{\mathrm{q}}$ and $\mathbf{b} = \mathbf{f}_{\mathrm{q}}^{t+1} - \mathbf{C}_{\mathrm{q}}\widetilde{\dot{\mathbf{q}}}^{t+1} - \mathbf{K}_{\mathrm{q}}\tilde{\mathbf{q}}^{t+1}$.

Due to the use of implicit time integration solver, the acceleration $\ddot{\mathbf{q}}$ within a single time-step is a constant. The unconstrained system is described by (3). When user interactions are applied, the corresponding spatial constraint force, described by (5), is treated as force feedback. If acceleration is lost, a predicted force can be computed based on the predicted acceleration. Note that two predictors, $\tilde{\mathbf{q}}^{t+1}$ and $\widetilde{\dot{\mathbf{q}}}^{t+1}$, can be computed directly using the physical status in previous time-step, so the value of $\mathbf{b}$ is available. The matrix $\mathbf{A}$ only depends on the properties of the 3D model under the linear elasticity assumption so it can be prestored on the receiver side when the system is initialized. Hence, following (10), force can be computed successfully on the user side. The computational cost of force simulation is very small. To ensure high-performance haptic response, we can employ a multithread solution. Alternatively, linear interpolation can be applied to ensure haptic rendering in a high frequency. Following the proposed force simulation algorithm, we can keep the consistency between visual and haptic feedback while successfully avoiding heavy haptic streaming and its related issues from unstable networks.

Network	Round-Trip Time (ms)			Bandwidth (Mb/s)	Lost Packets ratio
	Min.	Max.	Mean		
#1	181	683	196	2	9%
#2	50	187	72	4	6%
#3	15	78	36	10	3%
Local LAN	0.12	1.7	0.34	100	0

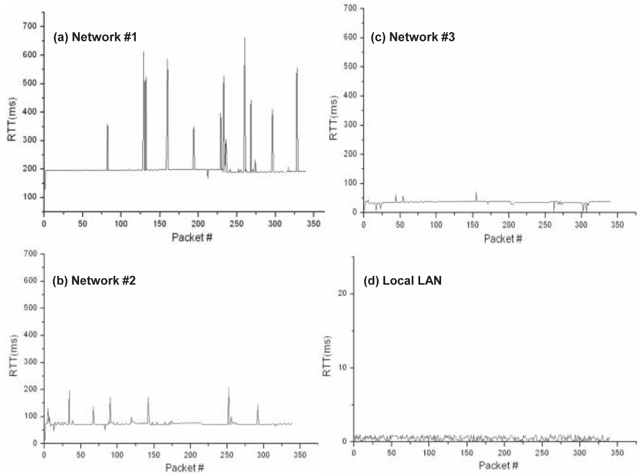


Fig. 6. Simulated four testing networks. Loss ratio of networks #1, #2, and #3 are 9, 6, and 3 percent, respectively. Bandwidth of networks #1, #2, and #3 are 2, 4, and 10 Mb/s, respectively.

6 Experimental Results

The proposed system and algorithms have been tested using various 3D models under different settings to evaluate the performance from different aspects. Below, we show some results and discuss our understanding.

6.1 Experimental Setting

The proposed collaborative virtual system is implemented on several Windows XP PCs with Intel Core2 Duo 2.93-GHz CPU and 2-GB DDR2 RAM. Two PHANTOM Omni devices and one PHANTOM Premium device are used as haptic hardware. HLAPI of OpenHaptics toolkit is employed to facilitate the force simulation and fill the gap of haptic rendering pipeline and graphics rendering loop. All the computers and haptic devices are in the same room during experiments and users communicate following the UDP protocol over different network conditions simulated with Dummynet [45]. Specifically, four simulated networks with various latencies, packet loss ratios, and bandwidths have been used for experiments, as depicted in Fig. 6. Network latency is computed as the round-trip time (RTT), for example, the mean latency in the network case #1 is 196 ms.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Model Statistic and Computational Time

Model	Model Statistics		Pre-computation Time (s)	Running Time (FPS)	
	#Tetra.	#Modes		Server	Client With/Without Compensation
Bar	1,080	20	0.11	1124.2	1124.2/1124.213
Bunny	50,000	60	14.7	264.2	264.17/264.25
Armadillo	49,729	120	21.3	134.8	134.76/134.78
Dragon	100,000	100	33.6	56.8	56.76/56.78

Bandwidth has been set as 2, 4, 10, and 100 Mbps with a loss ratio ranging from 0 to 9 percent. When a packet loss is detected, the NACK mechanism is applied to notify the sender to resend the lost data. The simulation time-step is set to be 1/40 second ($h = 1/40$).

6.2 Evaluating Compensation and Predition Algorithm

Table 2 reports the statistics of the testing models, their corresponding frequency mode, and computational time, including precomputation and running time. Precomputation refers to the procedure in which the Euler-Lagrange equation of the 3D model is converted into the spectral domain, i.e., (2), and it only needs to be performed offline once. As such, it has no influence on the system performance in terms of the running time, although the eigen-decomposition takes longer time to complete. The real-time performance is represented using the update frequency, i.e., frames per second (FPS). As shown in the table, we achieve fairly high update rates, even for very large models, implying that interactive deformation can always be ensured. Moreover, the computational cost of the proposed algorithm is also included in Table 2. It is clear that the performance with and without compensation algorithm is almost the same, indicating that the proposed algorithm is very efficient. Similarly, force simulation cost is also fairly small. Since force and deformation are causally related and computed concurrently, haptic update rate is the same as the simulation rate. To ensure a realistic and stable force, we adopt the OpenHaptics HLAPI with an internal low-level thread to execute haptic rendering at a higher rate (1 KHz).

To demonstrate the compensation and prediction results, we compare simulation with and without the proposed algorithm in Fig. 7, where random loss is involved. Four different cases are tested: (a) streaming without loss (black curve), (b) streaming with loss but no compensation and no prediction (red curve), (c) streaming with loss and compensation but no prediction (blue curve), and (d) streaming with loss, compensation, and prediction (green curve). In all four cases, displacement distortion is linear with respect to time, confirming our theoretic analysis in Section 5.3. Note that the prediction here refers to the prediction of acceleration (step 3) in the proposed loss compensation algorithm. The lost data can be set as zero or predicted to be $\hat{\ddot{\mathbf{q}}}^{\mathbf{i}}$ in (8) and (9). Therefore, compared to case (c), a linear prediction is applied in the case (d), so the simulation result is smooth and close to the no-loss case, as shown in Figs. 7c and 7d. We can also see from Fig. 7b that without applying the compensation the deformation errors accumulate along time: the distortion could be minor at the beginning but become very significant as the time passes. There are totally 22 random losses in the experiments. The first one occurs at t_{135}, and the first compensation happens at about t_{185}.

Force simulation results are included on the right side of each corresponding cases in Fig. 7. It is clear that the force feedback in all four cases is consistent with its corresponding deformation, which is determined by (5) or (10). Therefore, it ensures that users receive compatible visual and haptic feedback. For example, displacement curve in Fig. 7b shows that the model stops at a position different to its original one due to the data loss, and consequently the force is also nonzero. In addition, as force simulation is accomplished at users' side, network-related problems do not cause wrong perception, so that stable haptic feedback is guaranteed.

We test different compensation strategies and illustrate experimental results in Fig. 8. Similarly as in Fig. 7, there are

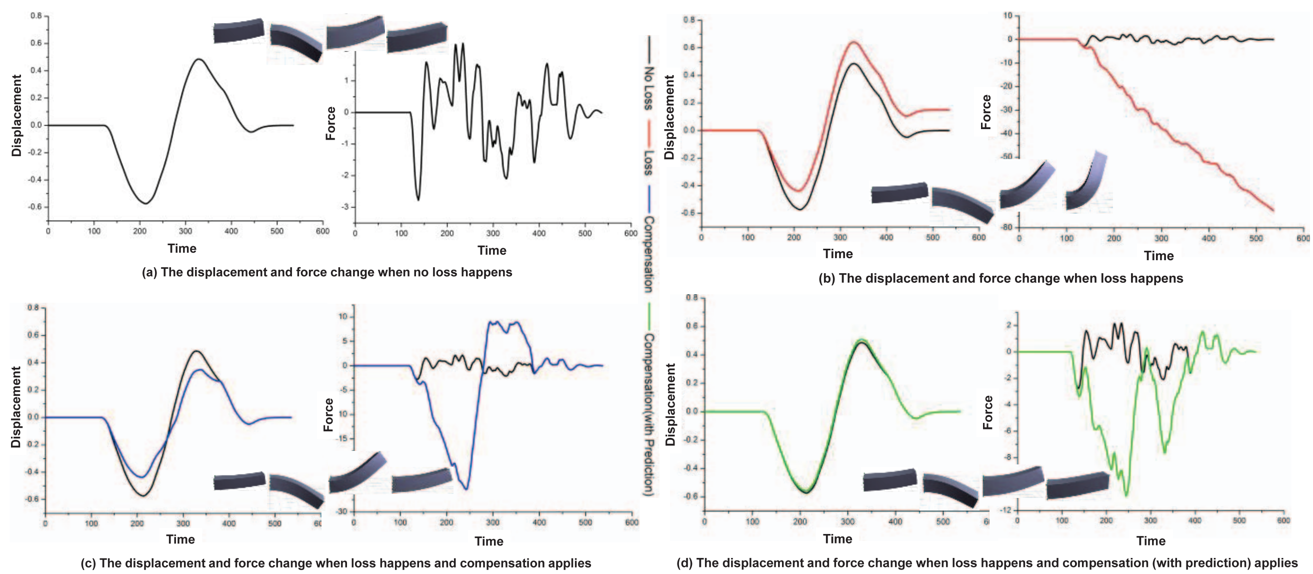


Fig. 7. Four different streaming results.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

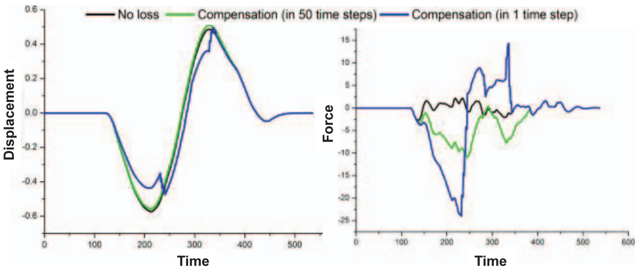


Fig. 8. Results of different compensation strategies.

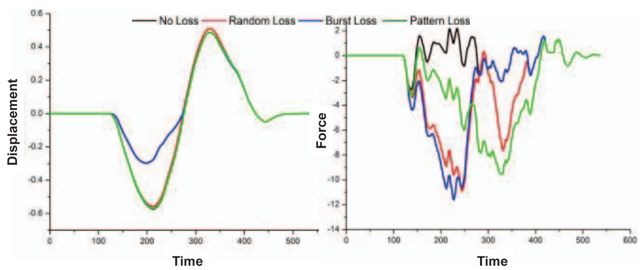


Fig. 9. Compensation results on different packet loss situations.

22 randomly lost data. As mentioned in Section 5.3, the step 5 of the proposed algorithm suggests that compensation can be completed in several time-steps instead of one to achieve a more natural and smoother deformation. To evaluate this, we compare two simulations cases: 1) compensation in one single time-step; and 2) compensation in 50 continuous time-steps. In the first case, depicted by the blue curve in Fig. 8, there are simulation jitters caused by sudden displacement and velocity changes. Similar jitter occurs on the haptics curve as well. The jitter is obvious if the time difference between packet loss and retransmitted data arrival is large, as a larger difference implies more accumulated errors. Consequently, compensating the large difference in one single time-step leads to the big change in simulation. Meanwhile, in the second case, as shown by the green curve of Fig. 8, when errors are gradually compensated in 50 time-steps, a more realistic and natural movement can be reached. The corresponding force is also smoother and stabler.

6.3 Evaluating the System Performance

The proposed system is tested in four different network cases, as shown in Fig. 6. The performance of all cases is satisfactory: users are able to visualize and feel the 3D model's motion in real time, and the system is consistent even with a high loss ratio. As explained before, the computational time used for compensation and prediction is negligible, so the network delay is a dominating factor which influences the system's synchronization. Based on the analysis in Section 3, we know that the size of streaming data also affects communication delay and loss ratio. In our HCVE, deformation update described by a few frequencies is a very small amount of data. When the first 20$\sim$100 modes ($m = 20 \sim 100$) are used for simulation, the streaming data size on both the C-S and P2P architectures is only a few KB, as shown in Table 3. It is clear that this small streaming data is very unlikely to cause frequent data loss, long delay, or serious jitter. Moreover, as force is computed on the receivers' side, we ensure stable, smooth, and deformation-consistent force feedback following the proposed force simulation algorithm.

TABLE 3
Communication Data Size

	Server	Client
Server	--	2~3KB
Client	1~2KB	--

(a). C-S

	Peer *i*	Peer *j*
Peer *i*	--	1~2KB
Peer *j*	1~2KB	--

(b). P2P

We have tested different network loss configurations to further evaluate the robustness of our system. Fig. 9 shows the testing results under three common packet loss situations, namely *random* loss, *burst* loss, and *pattern* loss. In the random loss case, packets are randomly lost during transmission, similar as in Figs. 7 and 8. Burst loss refers to a sequence of packet loss in a short time period. We use 12 consecutive losses in this experiment and the first loss happens at t_{140}. Pattern loss happens in a cyclic time period, for example, one packet loss every 50 time-steps. In this case, there are total 11 losses during pattern loss testing, and the first one happens at t_{135}. Compensation procedure is performed about 50 time-steps after each data loss, and is completed in 50 continuous steps. As shown in Fig. 9, the proposed algorithm is able to handle all three loss situations well, and generate smooth and natural results. Compensation results of random and pattern cases are very close to no-loss case, but the burst loss case (blue line) is a little less satisfactory. One reason could be that when all packets are lost in a short time period, predictions that are computed based on past few steps are not very accurate, and the prediction errors could also propagate and accumulate. We believe using a longer history to perform prediction should improve the result.

7 Conclusion and Discussion

In this paper, we have proposed a novel system to support real-time haptic interaction with physically based 3D deformable models in a distributed environment. We have followed the linear modal analysis and used the Modal Warping method to describe deformation using spectral accelerations, so that we can achieve a very small network load and low computational cost, and ultimately real-time streaming. To generalize our system, we have discussed how it works in two basic distributed topologies (C-S and P2P) over nondedicated networks where packet loss, delay, and jitter can happen. To tolerate errors caused by unstable networks, we have proposed a loss compensation and prediction algorithm to obtain smooth and physically correct movements. Our algorithm can be easily extended to other types of numerical time integration methods. Moreover, to ensure visual and haptic consistency and stability of force feedback, we have proposed a receiver side force simulation approach.

In the current research stage, our simulation is based on linear elasticity (e.g., with the use of Cauchy strain tensor) for the sake of computational simplicity. To handle large rotational deformation, the Modal Warping technique is adopted. It is worth noting that our system can also be

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

extended to simulate deformation using full Green tensor. In such case, the reduced stiffness matrix is not a constant and is varying with respect to the reduced displacement ($\mathbf{q}$). As introduced in [36], the element in the reduced stiffness matrix is a polynomial of the reduced displacement, and the coefficients of the polynomial can be precomputed. However, the time complexity of computing the 3D deformation at each time-step is $O(r^3)$, where r is the size of the deformation subspace (it also noted as the number of frequency mode in previous sections). Therefore, we think, the nonlinear modal analysis is not suitable for real-time simulation of deformable models with relative large size of subspace (i.e., over 50), which is also the reason why we did not choose to use nonlinear modal analysis for simulation.

On the other hand, the proposed loss compensation algorithm is based on the assumption of hyperelasticity, which means the shape of the object is not dependent on the deformation history. In many other cases, especially some biomechanic applications, the viscoelastic or history-dependent deformation is often required [40], [46]. Packet loss, delay, and jitter for such material behaviors lead to a profounder impact of the subsequent deformation of the object, as the lost data could contribute to future shape deviations. Thus, the deformation compensation in these cases will not be the same as our current one. Similarly, haptic simulation cannot be handled in the same way if viscoelastic or history-dependent material behaviors are involved. We believe extending our researches to handle network problems for viscoelastic materials is an interesting and challenging future work.

Another limitation of the proposed system is the system synchronization when permanent packet loss happens. Even though NACK is applied to notify the sender of data loss, NACK packet can be lost too and this could cause permanent data missing and system inconsistency. This limitation can be solved by performing periodically system check and data compensation if necessary. In this paper, our experiments focus on packet loss compensation and its related haptic simulation in the C-S architecture. Additional experiments regarding the P2P structure are needed, although the compensation strategy is similar. We also notice that it may not be necessary/possible to require all clients to have the same model resolutions as the server, so supporting multiresolution simulations in distributed HCVE is an interesting future work as well.

Applying the current system to actual real-world tasks for further testing is not completed now and will be included in the future research plan. We would like to experiment it through long geographical distance and test the system performance further. We also plan to extend the current framework to handle multiple objects in one environment, including rigid, deformable, and hybrid models. Additionally, exploring usability studies and integrating GPU to further improve the computational speed are also important future works.

REFERENCES

[1] M.A. Otaduy and M.C. Lin, "Sensation Preserving Simplification for Haptic Rendering," *ACM Trans. Graphics*, vol. 22, pp. 543-553, 2003.

[2] M.A. Otaduy, N. Jain, A. Sud, and M.C. Lin, "Haptic Rendering of Interaction between Textured Models," *Proc. IEEE Visualization Conf.*, pp. 297-304, 2004.

[3] R. Komerska and C. Ware, "Haptic Task Constraints for 3D Interaction," *Proc. IEEE Symp. Virtual Reality*, pp. 270-277, 2003.

[4] E.L. Sallnaes, K. Rassmus-Groehn, and C. Sjoestroem, "Supporting Presence in Collaborative Environments by Haptic Force Feedback," *ACM Trans. Computer-Human Interaction*, vol. 7, no. 4, pp. 461-476, 2000.

[5] C. Basdogan, C.-H. Ho, M.A. Shrinivasan, and M. Slater, "An Experimental Study on the Role of Touch in Shared Virtual Environments," *ACM Trans. Computer-Human Interaction*, vol. 7, no. 4, pp.443-460. 2000.

[6] G.C. Burdea, *Force and Touch Feedback for Virtual Reality.* John Wiley & Sons, 1996.

[7] R. Iglesias, S. Casado, T. Gutiérrez, A. García-Alonso, W. Yu, and A. Marshall, "Simultaneous Remote Haptic Collaboration for Assembling Tasks," *Multimedia Systems*, vol. 13, no. 4, pp. 263-274, 2008.

[8] J. Marsh, M. Glencross, S. Pettifer, and R. Hubbold, "A Network Architecture Supporting Consistent Rich Behavior in Collaborative Interactive Applications," *IEEE Trans. Visualization and Computer Graphics*, vol. 12, no. 3, pp. 21-10, May 2006.

[9] C. Gunn, M. Hutchins, D. Stevenson, M. Adcock, and P. Youngblood, "Using Collaborative Haptics in Remote Surgical Training," *Proc. Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 481-482, 2005.

[10] P. Buttolo, R. Oboe, and B. Hannaford, "Architectures for Shared Haptic Virtual Environments," *Computers and Graphics* vol. 21, pp. 421-429, 1997.

[11] X. Shen, F. Bogsanyi, L. Ni, and N.D. Georganas, "A Heterogeneous Scalabale Architecture for Collaborative Haptics Environments," *Proc. IEEE Int'l Workshop Haptic Audio and Visual Environments*, pp. 113-118, 2003.

[12] A. Pentland and J. Williams, "Good Vibrations: Modal Dynamics for Graphics and Animation," *ACM SIGGRAPH Computer Graphics*, vol. 23, no. 3, pp. 207-214, 1989.

[13] M.G. Choi and H.-S. Ko, "Modal Warping: Real-Time Simulation of Large Rotational Deformation and Manipulation," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 1, pp. 91-101, Jan. 2005.

[14] Z. Tang, Y. Yang, X. Guo, and B. Prabhakaran, "On Supporting Collaborative Haptic Interactions with Physically-Based 3D Deformations," *Proc. IEEE Int'l Workshop Haptic Audio and Visual Environments*, pp. 19-24. 2010.

[15] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically Deformable Models," *ACM Trans. Graphics*, vol. 21, no. 4, pp. 205-214, 1987.

[16] D. Terzopoulos and A. Witkin, "Physically Based Models with Rigid and Deformable Components," *IEEE Computer Graphics and Applications*, vol. 8, no. 6, pp. 41-51, Nov. 1988.

[17] D.L. James and D.K. Pai, "ArtDefo: Accurate Real-Time Deformable Objects," *Proc. ACM SIGGRAPH*, pp. 65-72, 1999.

[18] T.J. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Prentice Hall, 1987.

[19] N. Galoppo, M.A. Otaduy, P. Mecklenburg, M. Gross, and M.C. Lin, "Fast Simulation of Deformable Models in Contact Using Dynamic Deformation Textures," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, pp. 73-82, 2006.

[20] M. Hauth and W. Strasser, "Corotational Simulation of Deformable Solids," *J. WSCG*, vol. 12, nos. 1-3., pp. 137-145, 2003.

[21] J. Huang, Y. Tong, K. Zhou, H. Bao, and M. Desbrun, "Interactive Shape Interpolation through Controllable Dynamic Deformation," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 7, pp. 983-992, July 2011.

[22] F. Hecht, Y. Lee, J.R. Shewchuk, and J.F. O'Brien, "Updated Sparse Cholesky Factors for Corotational Elastodynamics," *ACM SIGGRAPH Trans. Graphics*, vol. 31, no. 5, article 123, 2012.

[23] K. Hauser, C. Shen, and J.F. O'Brien, "Interactive Deformation Using Modal Analysis with Constraints," *Proc. Graphics Interface*, pp. 247-255, 2003.

[24] C. Basdogan, "Real-Time Simulation of Dynamically Deformable Finite Element Models Using Modal Analysis and Spectral Lanczos Decomposition Methods" *Proc. Medicine Meets Virtual Reality Conf. (MMVR '01)*, pp. 46-52, 2001.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.

[25] D.L. James and D.K. Pai, "DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware," *ACM Trans. Graphics,* vol. 21, no. 3, pp. 582-585, 2002.

[26] N. Raghuvanshi, B. Lloyd, K.N. Govindaraju, and M. Lin, "Efficient Numerical Acoustic Simulation on Graphics Processors Using Adaptive Rectangular Decomposition," *IEEE Trans. Visualization and Computer Graphics,* vol. 15, no. 5, pp. 789-801, Sept./Oct. 2009.

[27] X. Guo and H. Qin, "Real-Time Meshless Deformation: Collision Detection and Deformable Objects," *Computer Animation and Virtual Worlds,* vol. 16, nos. 3/4, pp. 189-200, 2005.

[28] M.G. Choi, S.Y. Woo, and H-S. Ko, "Real-Time Simulation of Thin Shells," *Proc. Eurographics,* pp. 349-354, 2007.

[29] Y. Yang, G. Rong, L. Torres, and X. Guo, "Real-Time Hybrid Solid Simulation: Spectral Unification of Deformable and Rigid Materials," *Computer Animation and Virtual Worlds,* vol. 21, nos. 3/4, pp. 151-159, 2010.

[30] Z. Tang, G. Rong, X. Guo, and B. Prabhakaran, "Streaming 3D Shape Deformations in Collaborative Virtual Environment," *Proc. IEEE Virtual Reality Conf.,* pp. 183-186, 2010.

[31] G. Rong, Y. Cao, and X. Guo, "Spectral Mesh Deformation," *Visual Computer: Int'l J. Computer Graphics,* vol. 24, no. 7, pp. 787-796, July 2008.

[32] M. Bro-Nielsen and S. Cotin, "Real-Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation," *Computer Graphics Forum,* vol. 15, no. 3, pp. 57-66, 1996.

[33] S. Cotin, H. Delingette, and N. Ayache, "Real Time Elastic Deformations of Soft Tissues for Surgery Simulation," *IEEE Trans. Visualization and Computer Graphics,* vol. 5, no. 1, pp. 62-73, Jan.-Mar. 1999.

[34] D.L. James and D.K. Pai, "A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics," *Haptics-e, Electronic J. Haptics Research,* vol. 2, no. 1, Sept. 2001.

[35] S. Jun, J. Choi, and M. Cho, "Physics-Based S-Adaptive Haptic Simulation for Deformable Object," *Proc. Symp. Haptic Interfaces For Virtual Environment and Teleoperator Systems,* pp. 72-78, 2006.

[36] J. Barbič and D.L. James, "Six-DoF Haptic Rendering of Contact between Geometrically Complex Reduced Deformable Models," *IEEE Trans. Haptics,* vol. 1, no. 1, pp. 39-52, Jan.-Mar. 2008.

[37] R. Mafi, S. Mirouspour, B. Mahdavikhah, B. Moody, K. Elizeh, A.B. Kinsman, and N. Nicolici, "A Parallel Computing Platform for Real-Time Haptic Interaction with Deformable Bodies," *IEEE Trans. Haptics,* vol. 3, no. 3, pp. 211-223, July-Sept. 2010.

[38] J. Qin, K. Choi, and P. Heng, "Collaborative Simulation of Soft-Tissue Deformation for Virtual Surgery Applications," *J. Medical Systems,* vol. 34, pp. 367-378, 2010.

[39] S. Ullrich and T. Kuhlen, "Haptic Palpation for Medical Simulation in Virtual Environments," *IEEE Trans. Visualization and Computer Graphics,* vol. 18, no. 4, pp. 617-625, Apr. 2012.

[40] I. Peterlik, M. Sedef, C. Basdogan, and L. Matyska, "Real-Time Visio-Haptic Interaction with Static Soft Tissue Models Having Geometric and Material Nonlinearity" *Computers and Graphics,* vol. 34, no. 1, pp. 43-54, 2010.

[41] I. Fukuda, S. Matsumoto, M. Iijima, K. Hikichi, H. Morino, K. Sezaki, and Y. Yasuda, "A Robust System for Haptic Collaboration over the Network," *Touch in Virtual Environments: Haptics and the Design of Interactive System,* pp. 137-157, Pearson Education, 2002.

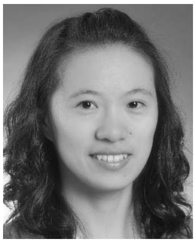
[42] J. Kim, H. Kim, M. Manivannan, M.A. Srinivasan, J. Jordan, J. Mortensen, M. Oliveira, and M. Slater, "Transatlantic Touch: A Study of Haptic Collaboration over Long Distance," *Presence: Teleoperators and Virtual Environments,* vol. 13, no. 3, pp. 328-337, 2004.

[43] H. Al Osman, M. Eid, R. Iglesias, and A. El Saddik, "ALPHAN: Application Layer Protocol for Haptic Networking," *Proc. IEEE Int'l Workshop Haptic Audio and Visual Environments,* pp. 96-101, 2007.

[44] N.M. Newmark, "A Method of Computation for Structural Dynamics," *J. Eng. Mechanics Division,* vol. 85, no. 3, 1959.

[45] L. Rizzo, "The Dummynet Project," http://info.iet.unipi.it/~luigi/dummynet/, 2011.

[46] M. Sedef, E. Samur, and C. Basdogan, "Real-Time Finite-Element Simulation of Linear Viscoelastic Tissue Behavior Based on Experimental Data," *IEEE Computer Graphics and Applications,* vol. 26, no. 6, pp. 58-68, Nov. 2006.



Ziying Tang received the PhD degree from the University of Texas at Dallas in 2011. She is currently an assistant professor at the Department of Computer and Information Sciences, Towson University, Baltimore, Maryland. Her current research interests include computer graphics, multimedia streaming, animation and visualization, physics-based modeling, haptics, and human-computer interaction.

Yin Yang received the PhD degree in computer science from the University of Texas at Dallas in 2013. He is an assistant professor in the Electrical Communication Engineering Department, University of New Mexico, Albuquerque. His research interests include physics-based animation/simulation and related applications, scientific visualization, and medical imaging analysis.

Xiaohu Guo received the PhD degree in computer science from the State University of New York at Stony Brook in 2006. He is an associate professor of computer science at the University of Texas at Dallas. His research interests include computer graphics, animation and visualization, with an emphasis on geometric- and physics-based modeling, spectral geometric analysis, deformable models, centroidal Voronoi tessellation, GPU algorithms, and 3D and 4D medical image analysis. He received the prestigious US National Science Foundation CAREER Award in 2012. He is a member of the IEEE.

Balakrishnan Prabhakaran received the PhD degree in computer science from the Indian Institute of Technology, Madras, India, in 1995. He is currently a professor of computer science in the University of Texas at Dallas. He has been working in the area of multimedia systems: animation and multimedia databases, authoring and presentation, resource management, and scalable web-based multimedia presentation servers. He received the US National Science Foundation CAREER Award in 2003. He was the general cochair for the ACM Multimedia 2011 and has served as an associate chair of the ACM Multimedia Conferences in 2006, 2003, 2000, and 1999. He has served as the guest editor of special issues on various topics for different multimedia journals. He also serves on the editorial board of journals such as *Multimedia Systems* (Springer), *Multimedia Tools and Applications*(Springer), *the Journal of Multimedia* (Academy Publishers), and the *Journal of Multimedia Data Engineering and Management* (Information Resources Management Association). He is also the editor-in-chief of the *ACM SIGMM* online magazine.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: CLEMSON UNIVERSITY. Downloaded on March 30,2021 at 01:19:10 UTC from IEEE Xplore. Restrictions apply.