SCeVE: A Component-based Framework to Author Mixed Reality Tours

SHANTHI VELLINGIRI, University of Texas at Dallas, USA
RYAN P. MCMAHAN, University of Central Florida, USA
BALAKRISHNAN PRABHAKARAN, University of Texas at Dallas, USA

Authoring a collaborative, interactive Mixed Reality (MR) tour requires flexible design and development of various software modules for tasks such as managing geographically distributed participants, adaptable travel and virtual camera techniques, data logging for assessment of the incorporated techniques, as well as for evaluating the Quality of Experiences (QoE). In most cases, authors might have to develop all these software modules, instead of focusing only on the virtual environment design. In this article, we propose SCeVE, a component-based framework that supports flexible design and authoring of interactive MR tours by offering ease of access to four major design choices: (i) **S**ynchronization, (ii) **C**ollaborative **e**xploration, (iii) **V**isualization, and (iv) **E**valuation. Based on tour requirements, an author can access one or more *components (or software libraries)* of design choices via *SCeVE's API (Application Programming Interface) services*, as demonstrated by the two case studies on group travel in a plant walk MR tour.

SCeVE framework is innovative in the sense that it facilitates group travel in virtual environments involving "live" models of participants from geographically distributed sites. SCeVE empowers authors to focus only on the design of the required virtual environments. They can quickly build a diverse set of collaborative MR tours by utilizing the flexibility of SCeVE in terms of the various available options for traveling, rendering on multiple devices, and virtual camera viewpoint computation strategies. By providing data logs of various components, SCeVE facilitates performance evaluation of the various strategies used as well as the user experience in collaborative MR tours. SCeVE is designed in an extensible manner, allowing authors to add devices and software services as additional components.

CCS Concepts: • **Human-centered computing → Gestural input**; **Interaction design theory, concepts and paradigms**; • **Computing methodologies → Modeling methodologies**; • **Computer systems organization → Distributed architectures**;

Additional Key Words and Phrases: Authoring, virtual reality, mixed reality, trave, gesture-based, non-natural, framework, components, services, case study, modeling, user experience

This material is based upon work supported by the US Army Research Office (ARO) Grant W911NF-17-1-0299. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ARO.

Authors' addresses: S. Vellingiri, University of Texas at Dallas, 800 W. Campbell Road, Richardson, Texas, 75080-3021; email: shanthi.vellingiri@utdallas.edu; R. P. McMahan, University of Central Florida, 4000 Central Florida Blvd., Orlando, Florida; email: rpm@ucf.edu; B. Prabhakaran, University of Texas at Dallas, 800 W. Campbell Road, Richardson, Texas, 75080-3021; email: bprabhakaran@utdallas.edu.

1551-6857/2020/05-ART40 $15.00
https://doi.org/10.1145/3377353

ACM Reference format:
Shanthi Vellingiri, Ryan P. Mcmahan, and Balakrishnan Prabhakaran. 2020. SCeVE: A Component-based Framework to Author Mixed Reality Tours. *ACM Trans. Multimedia Comput. Commun. Appl.* 16, 2, Article 40 (May 2020), 23 pages.
https://doi.org/10.1145/3377353

1 INTRODUCTION

Consider a museum or a botanical walk showcasing exhibits or objects-of-interest. Communication among the visitors assists them in determining where to halt, how to take turns in listening to the audio and video narration of an exhibit, accommodate other visitors, and so on. A virtual replica of the above-mentioned scenario requires collaboration, travel, interaction, communication, and viewpoint control [47, 51, 61] for effective experience.

A Mixed Reality (MR) tour involves various physical sites (i.e., local and remote). A site involves one to several participants and several devices to assist content generation, content visualization, interaction detection, and so on. A research study analyzing an aspect of a tour requires a custom design of the tour. For instance, a study considers the advantages of a collaborative guided tour involving gesture-based techniques for travel, whereas another study analyzes the impacts of an unguided tour. As research studies focus on improving the functionalities of an existing MR tour, the design and development of it must be less time-consuming. Therefore, a framework that accommodates and manages available (or additional) resources (input/output devices, libraries) to support expeditious authoring to enhance the potential of MR tours is essential.

1.1 SCeVE Framework

Though there exist many component-based frameworks that support the design of Virtual Reality (VR)/Mixed Reality (MR)/Augmented Reality (AR) applications, those were built to accommodate assembling, manipulating, information seeking, and storytelling tasks. Most of them lack supporting "live" model-based collaboration, group travel using real-world interactions, flexible visualization, and viewpoint computation strategies. The SCeVE framework offers flexibility to accommodate another research problem of the related genre by providing: (i) collaboration among the participating sites, (ii) efficient data logging and management, (iii) quick integration of assessment models, and (iv) fast development phase. SCeVE is a component-based framework (shown in Figure 1) and named after its potential to offer the key design choices: (i) **S**ynchronization, (ii) **C**ollaborative **e**xploration, (iii) **V**isualization, and (iv) **E**valuation.

- `Synchronization`:
  - The primary objective of a tour is to educate users on exhibits and their specialties, and interactive narration requires multimedia elements such as text, audio, and images. Synchronization [62, 66] among various multimedia elements is inevitable to provide a meaningful narration about an exhibit and its features. Findings of Reference [56] suggest including narration on exhibits enhances a museum visitor's experience.
  - In-addition, when a "live" [21, 45, 64] model is preferred, synchronization of several data frames from a content generation device is essential to enhance user experience. For instance, using a Kinect camera for content generation, a user's virtual model is generated by synchronizing [60] the color, depth, and skeleton data streams of Kinect. For instance, as a user performs a hand-waving gesture, his/her corresponding "live" model has to mimic the same gesture with very minimal delay.

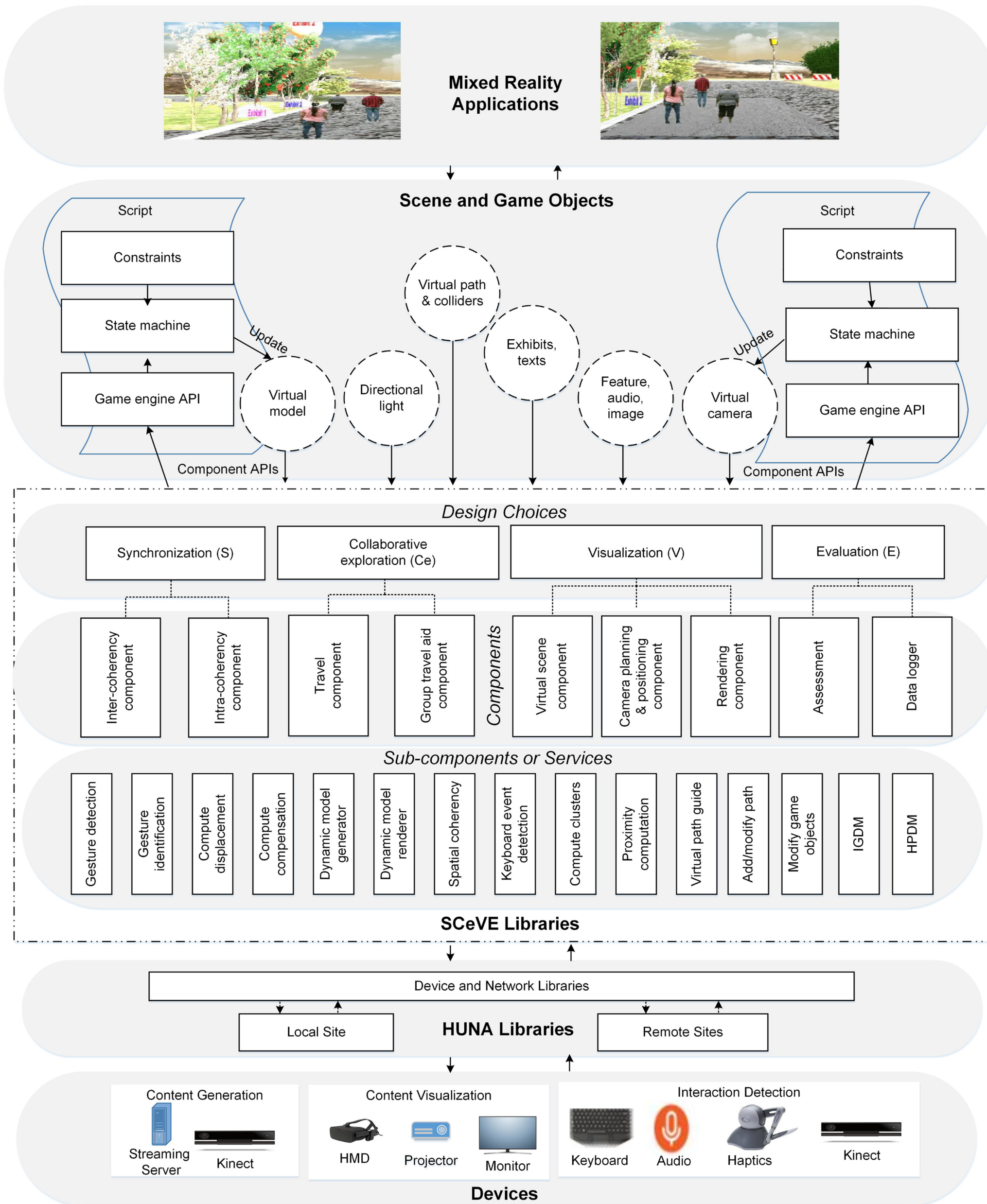

Fig. 1. High-level architecture of SCeVE framework.

- `Collaborative exploration:`
  - Collaboration among the participating sites supports task completion and improves the user experience. Passive, semi-active, and active task types distinguish the nature of a collaboration task. Participants can choose a coequal or a leader-follower role to complete the assigned task. Based on the chosen roles, participants may seek a guided or unguided tour. For instance, in a guided virtual tour, as a guide is familiar with tour aspects,

such as tour path, starting location, finish location, exhibits, and their details, he/she can organize a virtual tour by notifying other participants on virtual exhibits locations, their significance, and so on. Collaboration among participants is essential in a guided tour. Whereas, in an unguided tour, participants solely decide on their travel and exhibit preferences.
  — Travel choices among participants differ—one could opt for a natural travel technique such as real walking [51], semi-natural Walking-in-Place (WIP) [34], Arm Swinging (AS) [17], Tapping-in-Place (TIP) [52]), or non-natural (keyboard-based [27], speech-based [63], handheld-based [50]) travel technique. These travel techniques facilitate the movement of a user's virtual model ("live" or avatar-based) in the virtual environment.
- `Visualization`:
  — Visual effectiveness of an MR tour depends on its virtual camera planning and placement strategy (otherwise known as viewpoint control [61]). Potential camera planning and placement strategies [4, 10, 16, 42] are first-person, third-person, and action replay. Flexible virtual camera planning and positioning strategies are essential to improve a tour taking experience. For instance, as a user is near an exhibit, the movement of the camera towards the exhibit improves the learning aspect of the tour.
- `Evaluation`:
  — The effectiveness of collaborative exploration, virtual camera viewpoint computation strategies, learning experience, usability, Quality of Experience (QoE), and so on, requires subjective [13, 33, 37, 65, 69] or objective [31, 72, 73] assessment approaches. These methods are application-specific and need efficient data logging and retrieval services.

Each design choice consists of one or more *components (software libraries)* with various *services (computation, device access, database access)*. Based on an MR tour's requirement, an author of the tour can select appropriate design choices, components, and services to access some or all of the functionalities of SCeVE. A service typically involves *read* and *write* calls from entities (or game objects) of a virtual scene to a design choice to request access to an associated device for content generation, content visualization, or interaction detection. Section 3 discusses design choices, components, services, and network libraries of SCeVE.

1.2 Contributions

In this article, we describe the design of an innovative, collaborative MR tour authoring framework, SCeVE, with the following contributions:

- Authors can focus only on the design of the required virtual environments and build a diverse set of collaborative MR tours by using the flexibility of SCeVE in terms of the various available options for traveling, rendering on multiple devices, and virtual camera viewpoint computation strategies.
- Collaborative MR tours authored using SCeVE allow "live" models of participants from geographically distributed sites, thereby providing the tour users a more engaging and interactive experience with their collaborating peers.
- The data logging features in SCeVE use NTP (Network Time Protocol) [48] to synchronize the clocks of the collaborating sites. Hence, authors can conduct different research tasks such as UX (User Experience) studies or network/system performance-related studies very easily by using SCeVE's data logging features in each collaborating site.
- SCeVE is designed in an extensible manner, allowing authors to add devices and software services as additional components.

1.3 Article Organization

Here, we detail the organization of this article. Section 2 details the high-level architecture of SCeVE framework. Section 3 presents the design choices, components, device, and network libraries of the SCeVE framework. Section 4 discusses authoring of two travel-related studies. The studies evaluate the preferred travel technique that facilitates group travel experience. We define group travel as the virtual travel of users from multiple physical locations. While Reference [74] discusses the details of the travel techniques, research problem, associated analysis, and inference, in this article, we describe how the collaborative plant walk MR tour was authored using SCeVE. Section 5 compares SCeVE with existing VR/MR authoring frameworks. Section 6 provides the conclusion of the SCeVE framework.

2 HIGH-LEVEL ARCHITECTURE OF SCEVE FRAMEWORK

The SCeVE framework (shown in Figure 1) is a library to support the custom design of MR tours. The backbone of SCeVE is the various design choices that aid custom MR tour design. In this section, we discuss how we use SCeVE to author an MR tour from a game object's (an entity of a virtual scene or environment) point-of-view.

2.1 Virtual Scene with Game Objects

Scene creation is the foremost step in authoring an MR tour. Some of the popular MR scene authoring tools (or game engines) are Unity3D [70], Amazon Sumerian [5], and Unreal Engine [71]. Typically, a virtual scene is a collection of entities (game objects or assets) that support a tour. For instance, an MR tour scene consists of virtual entities such as (i) path, (ii) users model of all participating sites, (iii) exhibits or assets, (iv) camera, (v) lighting, (vi) canvas, (vii) audio objects, (viii) panels, (ix) images, and so on.

Generally, two types of game objects exist: (i) active game objects and (ii) supporting game objects. Active game objects remain dynamic and replicate the real-world actions of a user or participant. Supporting game objects are immobile and are essential to the building of a virtual scene. In an MR tour, synthetic avatars or "live" models of participants are some of the active game objects. Virtual exhibits, pathways, and panels are some of the supporting game objects.

Game objects perform a task based on its assigned characteristics. Typical characteristics are (i) physical property, (ii) physical appearance, (iii) ownership [36, 61]—defines whether a local or a remote site controls the game object in terms of device to access, (iv) design choice—required functionalities to satisfy, and (v) constraints—allowed behavior to perform a task.

While authoring and taking into consideration the nature of the game objects, an author assigns scripts (a program instructing a task including constraints to control its behavior) to game objects. A script controls a game object's behavior and aids in task completion. It involves (i) characteristics as public variables, (ii) API calls to framework libraries and authoring game engine tool, and (iii) a finite state machine (FSM) [38, 79] to facilitate the game object's transition.

2.2 Game Object and Component Access

A game object requires access to one or more design choices to perform and complete an assigned task. A design choice is a collection of related software components that serve a specific aspect of a tour. SCeVE (shown in Figure 1) offers synchronization, collaborative exploration, visualization, and evaluation design choices to author various MR tours. A design choice consists of one or more components or software libraries. These software libraries provide various services to exchange information from the device and network layers to the application layer (a virtual scene with game objects) and vice versa.

For instance, travel and group travel aid components of exploration design choice consists of several services. To mention a few: (i) *getFwdInteraction (type: device access, parameters)*, (ii) *proximityToExhibits (type: computation, parameters)*, and (iii) *dataLoggerTravel (type: database access, parameters)*. A game object can access the services of a component library through *read/write* API calls. A read or write call from a game object instance to a component library allows access to one or more services of that component. Following are some of the services that a dynamic game object (gO) instance can access:

- `Computation Service`:

$$gO = read/write(component(1...c).service(1...s), parameters(1...p)). \tag{1}$$

- `Device Access Service`:

$$\begin{aligned} gO = read/write(component(1...c).service(1...s), parameters(1...p), \\ device(1...d), streams(1...s), port(1...p)). \end{aligned} \tag{2}$$

- `Database Access Service`:

$$\begin{aligned} gO = read/write(component(1...c).service(1...s), \text{“}databasePath\text{”}, \\ dbTable(1...d), dBQuery(1...q)). \end{aligned} \tag{3}$$

2.3 State Transition of Game Object

As a game object performs a *read* API call to any of the components, its finite state machine is invoked to determine its state transition. The author of an MR tour decides possible states of a game object based on the role of the game object in the tour. Example states for a dynamic/active game object that represents a participant are: (i) *idle* state; (ii) *travel* state—enables travel on receiving interaction within a threshold; (iii) *compensate* state—reduce travel speed when game object is bound to collide with other game objects, or on proximity to virtual exhibits; (iv) *viewing* state—where game object can be closer to an exhibit and ready to look at the interactive displays about the exhibit; (v) *render* state—enables to visualize rendering of virtual scene and game objects. A detailed explanation of component calls and state transitions from a game object's point of view is presented in Figure 7 and discussed in Section 4.1.1.

For instance, consider a game object that is associated with a travel component issues *read* and obtains the interaction in virtual units. Later, the state machine on this game object evaluates whether this interaction unit is within a threshold (which is mentioned as a constraint of the game object), in which case, the game object's state is changed from *idle* to *travel*, and corresponding virtual displacement is updated to the game object, otherwise not. In this way, the behavior of the game object is maintained to achieve the assigned task.

In a virtual scene constituting 'n' game objects, based on the associated characteristics in their scripts, game objects ($i \leq n$) access SCeVE services and update their states, resulting in a virtual scene transition. A content visualization device (Head-Mounted Displays (HMDs), TV, projector) render these scene changes and aid visualization.

3 DESIGN DETAILS OF SCEVE FRAMEWORK

The high-level architecture of SCeVE is shown in Figure 1. The component libraries are implemented in C#. SCeVE is designed using a Unity3D [70] game engine. In this section, we discuss the components, its sub-components or services, device, and network libraries of SCeVE.

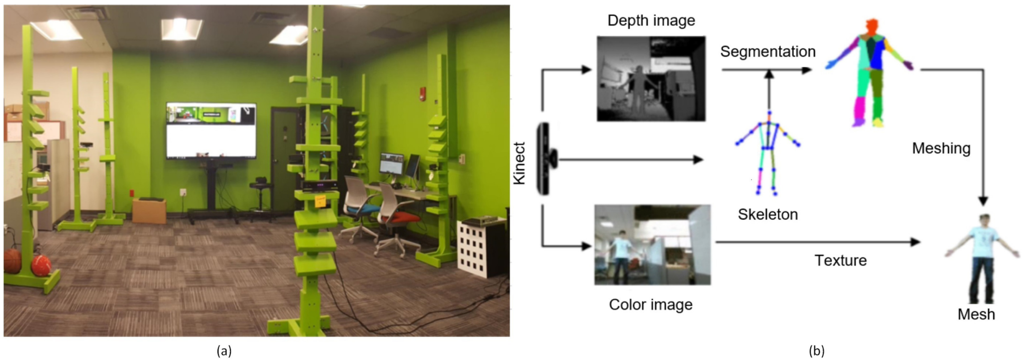

Fig. 2. (a) Capture space with several Kinect sensors, (b) Intra-coherency between of skeleton, depth, and color images of Kinect.

3.1 Design Choice for Synchronization

Synchronized rendering of "live" models and static game objects are essential to provide interactivity and enrich the user experience. NTP is used to synchronize the clocks of all the machines of the participating sites with millisecond precision.

3.1.1 Intra-coherency Component. Intra-media synchronization (shown in Figure 2) aids in constructing a "live" model of a participant. SCeVE framework uses Heterogeneous Ubiquitous Networked Architecture (HUNA) libraries [60] to obtain avatar content creation. Multiple calibrated RGB-D cameras capture a participant, reconstruct a 3D model of the participant, and transmit it to other collaborating sites—every frame. Figure 2 shows the capture space with several Kinect sensors. Calibration of RGB-D Kinect cameras at participating sites is through a skeleton-based, extrinsic calibration approach [22]. On receiving skeleton, depth, texture, and color frames from Kinect sensors, the surface mesh of a participant is created and rendered in real-time. This component aids in content creation and synchronizes the real-world action of a participant on to his/her "live" model to enhance body ownership [36, 61] rather than following a standard or readily available avatar.

3.1.2 Inter-coherency Component. Inter-media synchronization (shown in Figure 3) supports exhibit narration by timely coordination of multiple media elements such as text, audio, and images. Synchronization among image (a visual representation of the feature of an exhibit), its corresponding text, and audio narration not only maintain intact user interest but also improve the learning experience.

3.2 Design Choice for Collaborative Exploration

This design choice enables and controls travel (of a participating user and/or a group of users) in a virtual world. Based on a site's choice on travel devices and an MR tour's requirements, an end-user can opt for any of the components and its services. Figure 4 shows the components (and its services) of the collaborative exploration design choice.

3.2.1 Travel Component. It is one of the primary components of MR tour exploration. Based on a chosen travel technique, this component captures the real-world interaction of a participant and computes his/her virtual displacement. Popular interaction techniques [1, 18, 53] that aid travel in an MR tour are:

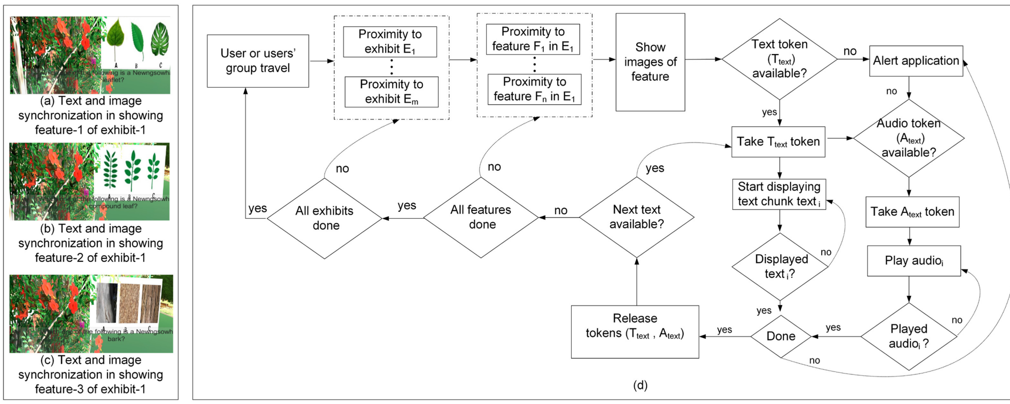

Fig. 3. (a) Shows a single leaf feature of a Newngsowha (exhibit-1) tree, (b) shows a compound leaf feature of a Newngsowha tree, (c) shows a bark feature of a Newngsowha tree, (d) shows the algorithm for inter-coherency between text, audio, and display images for each feature of plant walk exhibits.

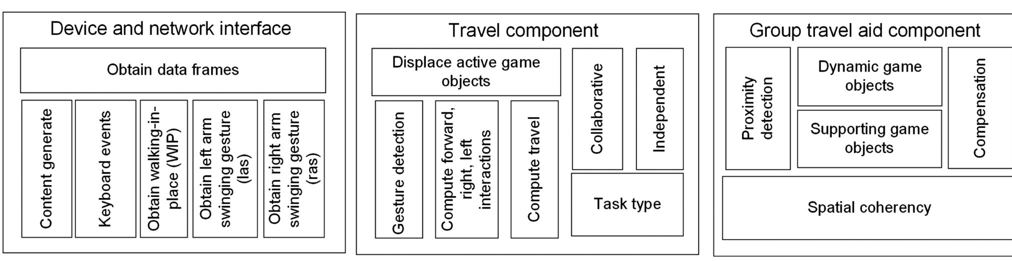

Fig. 4. Travel and group travel aid components (and their services) of collaborative exploration design choice.

- `Semi-natural Travel Technique`: The real-world gesture of a participating user is the steering or propelling force to navigate his/her synthetic or "live" model in a virtual world. Travel techniques such as Walking-in-Place (WIP) [34], Arm Swinging (AS) [17], and Tapping-in-Place (TIP) [52] are some of the examples.
- `Non-natural Travel Technique`: The steering force to displace an avatar or a "live" model is an event from an external device of choices such as a handheld, keyboard, or speech-based interface.

Irrespective of the nature of a travel technique, a user can take a tour independently or collaboratively. In either case, the travel component aids in computing the real-world interaction and the corresponding virtual displacement. Consider a participating user (say, denoted in virtual space as game object *i*) that follows semi-natural travel technique (WIP for forward displacement) to independently explore an MR tour. This game object instance can access *getFwdInteraction(gO_i, "wip")* service of the travel component to obtain the raw interaction and later invoke *computeDisplacement(fwd, none, none)* of the same component to obtain the actual displacement in the forward direction. Now, taking into account the constraint specified on this game object, its state machine verifies for a state transition and then allows the update of the virtual displacement on this game object *i* using *transform.Translate* Unity3D API. Allowing the game object *i* to virtually displace in the forward direction in the virtual environment.

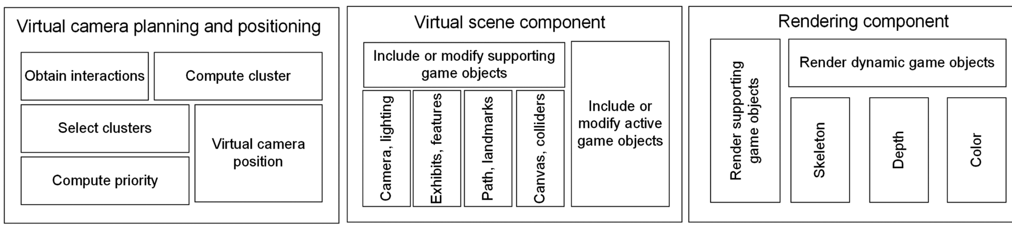

Fig. 5. Components and services of visualization design choice.

3.2.2 Group Travel Aid Component. This component regulates virtual world travel. Uncontrolled travel results in passing through [46, 68] among virtual models. Maintaining a minimum separation distance between participants' virtual models minimizes a virtual collision. Irrespective of a travel technique, each participant travels at a different rate. Group travel aid controls speed-up or slow-down of participants' "live" models and minimizes pass-through between them. As network and operating system latencies affect the collaboration, spatial synchronization algorithms such as averaging "live" models' 3D positions improve spatial coherency [73] among the participating sites.

3.3 Design Choice for Visualization

Visualizing a virtual scene and its contents based on a tour-taking user's interaction helps the user to immerse in an MR tour. Figure 5 shows the three components (and their major sub-components) of the visualization design choice.

3.3.1 Virtual Scene Component. Using Unity3D API, the virtual scene component allows additions and modifications of active (or dynamic) and supporting (or static) game objects to obtain a virtual scene. For instance, extending virtual pathways in its size and direction, including or replacing an existing virtual exhibit, changing start and end locations of an MR tour, and so on, are essential to modify an existing virtual scene to support another study.

3.3.2 Virtual Camera Planning and Positioning Component. A first-person camera planning and positioning algorithm is essential when participants choose independent exploration; whereas, collaboration requires a third-person camera planning and positioning algorithm. When users or a group of users travel in the virtual environment, as they approach an exhibit, the virtual camera shifts its preference to the exhibit and gradually advances towards the exhibit to showcase its specialties. Approaches such as the priority switching (shown in Figure 6) camera planning strategy can improve the educational aspect of the tour.

3.3.3 Rendering Component. Unity3D API facilitates the rendering of a virtual scene with active and supporting game objects. Until tour completion, the virtual scene and its contents are rendered on the selected content visualization device.

3.4 Design Choice for Evaluation

Mathematical model [31, 72, 73] is one of the common objective evaluation strategies. Mean Opinion Score (MOS) [13, 65, 69] is a popular subjective evaluation strategy. For either strategy, based on the tour parameters to assess, a suitable data collection or a data logger assists evaluation in data storage and retrieval.

3.4.1 Database Component. We chose a record-based data model of relational type to assist evaluation or assessment. The majority components of SCeVE require database services to aid

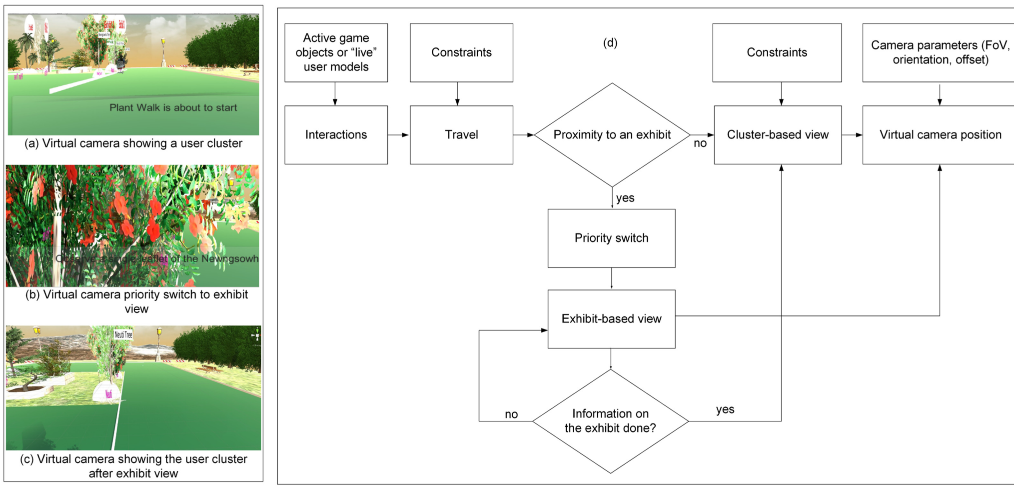

Fig. 6. (a) Shows the virtual camera behind a user, (b) shows the virtual camera focusing an exhibit as a user is near the exhibit, (c) shows the virtual camera behind the user after viewing the exhibit, (d) shows the priority switch virtual camera planning algorithm.

in storing and retrieving from corresponding database tables. For instance, as participants interact, their 3D position, amount of interaction, interaction direction, their neighbors, proximity to exhibits, voting for quizzes on exhibits, current virtual path, and so on, help assessment.

3.4.2 Assessment Component. This component includes heuristics-based models that evaluate MR tour aspects based on the requirements of a research study. For instance, the current version of SCeVE includes: (i) Hierarchical Position Discrepancy Model (HPDM) [73] and (ii) Inter-group Distance Model (IGDM).

- `HPDM`: Network delay among the participating sites can introduce intermittent travel and/or static visual rendering of the participating virtual models. As a result, a participant might notice a spread-out virtual group if they have opted for collaborative exploration, delayed rendering of virtual models of other sites in a participant's site. In a collaborative tour, these can degrade QoE. The assessment component provides services to access a heuristics-based model known as HPDM to assess QoE of a collaborative MR tour. The model takes a virtual position of dynamic game objects (or participants' "live" models) as input and obtains the amount of spatial incoherency in network delay scenarios. Later, the model uses spatial incoherency and users' scores to obtain QoE.
- `IGDM`: This model calculates travel speed of users. It also provides an inter-personal distance of each participating user with every other user in a collaborative tour and obtains their inter-group distance until tour completion. The inter-group distance provides the travel behavior of a group. A larger inter-group distance signifies that the group did travel apart and vice versa.

3.5 Device and Network Access

SCeVE access device and network libraries from the existing framework are called Heterogeneous Ubiquitous Networked Architecture (HUNA) [60]. The backbone of HUNA is the following four modules: (i) With a producer-consumer model, the *Device Module* obtains frames from the local and remote devices, organizes, and transfers the data frames in a predetermined format required to the other modules; (ii) Using the device SDK, the *Algorithm Module* offers background subtraction, region-based filtering, surface reconstruction, meshing, and assist content generation; (iii) *Streaming Module* composing network libraries establish a TCP/IP connection between local and remote sites based on a peer-to-peer (P2P) architecture; and (iv) *Renderer Module* facilitates the rendering through 3D game engines such as Unity3D and OpenGL.

The SCeVE framework supports Kinect RGB-D cameras (avatar creation or content generation), keyboard interface to facilitate non-natural travel technique, haptics [75] to provide force feedback, and HMD and projector displays for content visualization. Now, when a service of a component of SCeVE requires device access, it accesses the corresponding device module/library. Information on the type of input from the specific device library is obtained from the characteristics mentioned in the game object that needs the service. For instance, if a dynamic game object accesses a travel component to obtain device access service *getFwdInteraction*, this service will access the gesture detection class of the Kinect library to obtain the raw gesture. The game object, upon receiving the raw gesture, accesses computation service *computeDisplacement* to obtain displacement and invokes its state machine to check on its state transition. Upon success, the game object access database accesses *dataLoggerComponent(path, dbTable, parameters)* to store the displacement for time instant *t*. A similar process is followed for all the dynamic game objects at all sites. As the game objects transition, their corresponding virtual displacements are visualized through a content visualization device such as HMD or projector.

4 USING THE SCEVE FRAMEWORK

We use SCeVE to investigate the QoE of travel-based research problems. For instance, consider the exploration design choice: Several travel techniques support the exploration of an MR tour, but which travel technique can offer the best collaboration? Similarly, each design choice and combination of those can be used to study the potential of travel to an MR tour. To customize an MR tour, an author has to decide on the following:

- Changes to the physical properties of one or many game objects and/or scenes.
- Total physical sites and participants at each site. Number of active or dynamic game objects, their characteristics, constraints, and design choices.
- Start and finish locations of the tour. The number of exhibits to view multimedia resources for interactive narration. Addition or replacement of exhibits and/or features. Content generation, interaction detection, and content visualization devices.
- Preferred interaction type for forward, left, right travel directions to explore the tour. Selection of a suitable virtual camera planning algorithm. Select a local drive location and decide on the database type, data tables, entities to log.
- In case the above modifications require additions to the device or component library of SCeVE, identify and add appropriate inclusions in the corresponding component or device library.

4.1 Collaborative Plant Walk Using SCeVE

In this section, we discuss the authoring of a collaborative plant walk MR using SCeVE. Using the MR tour, we evaluate the potential of semi-natural and non-natural travel techniques to offer a better walking-as-a-group [74] tour experience. We define walking-as-a-group as traveling together with group members to collaboratively explore a tour. This can be achieved using any travel technique. However, we evaluate only semi-natural and non-natural travel techniques. These travel techniques have their pros and cons. For instance, a gesture-based semi-natural travel technique provides a more natural means of interacting with the system, while a non-natural travel technique requires less physical effort. We have used SCeVE to implement these types of techniques and their differences and to conduct a usability study comparing the effectiveness of these techniques under different network conditions.

4.1.1 Tasks. In this section, we describe how the collaborative plant walk tour for the semi-natural and non-natural travel techniques was authored using SCeVE. The link to a demo of the collaborative plant walk tour using a semi-natural travel technique is provided in Reference [20]. In our first case study, we authored an MR tour set up to support the semi-natural travel technique; and in our second case study, we authored an MR tour set up to support the non-natural travel technique.

- `Semi-natural`: For the semi-natural technique, we chose to use Walking-in-Place (WIP) [34] for travel in the forward direction, and Arm Swinging (AS) [17] provides right and left travel. For WIP, the steering force to navigate a participant's "live" model (or a virtual model) is a marching gesture. For AS, the user can adjust the direction of travel by swinging his/her arms left or right.
- `Non-natural`: For the non-natural travel technique, we chose to use keyboard-based input [39]. For the keyboard technique, the steering force is an event generated by pressing the directional keys of a keyboard. At a site, the "live" models of participating users are collocated (virtually) on a virtual object such as a virtual carpet. Each event displaces the virtual carpet by a constant of 10 units. Therefore, the virtual carpet carrying the participants' models moves at a constant, predetermined speed while the participants get a tour of the virtual world.

Case Study on the Semi-natural Travel Technique. Here, we discuss how we used SCeVE to create the semi-natural travel technique using gesture-based interaction. The design process that we followed (shown in Figure 7) is discussed below:

- We involve three dynamic or active game objects to represent the participant's "live" model and duplicate his/her real-world actions.
- We assign the following characteristics as a script on each of the dynamic game objects:

  game object [i] <local/remote, dynamic, projector, fwdInteration = "wip", rightInteraction = "ras", leftInteraction = "las", "dataPath", components(all), constraints(speed, threshold, separationDist, proximityDist, exhibitCloseness, cameraOffset, refreshTime)>

  Where, game object-i represents a participant's "live" model. In this case, the keyword "local" indicates that a local site owns this game object, the game object is dynamic, and it can access components and devices of the local site. Projector and Kinect are specified for content visualization and content generation, respectively.
- Dynamic game objects are instantiated and now based on assigned characteristics of the framework libraries.
- To include gesture-based (WIP, AS) travel techniques, game object[i] issues call to the following methods of travel component: *getFwdInteraction*, *getRightInteraction*, *getLeftInteraction*. These methods in turn call corresponding services of the gesture detection class, which in turn accesses the Kinect device class and calculates the raw displacement corresponding to the gestures. On receiving the interaction, the game object invokes the *computeDisplacement* service, which provides the game object[i] with the virtual world displacement corresponding to its participant's real-world interaction. Later, game object[i] invokes its state machine to verify whether the obtained displacement is within its threshold as constraints defined in its characteristics. Upon success, it decides on its transitions to travel and then displaces "x" units in the virtual world.

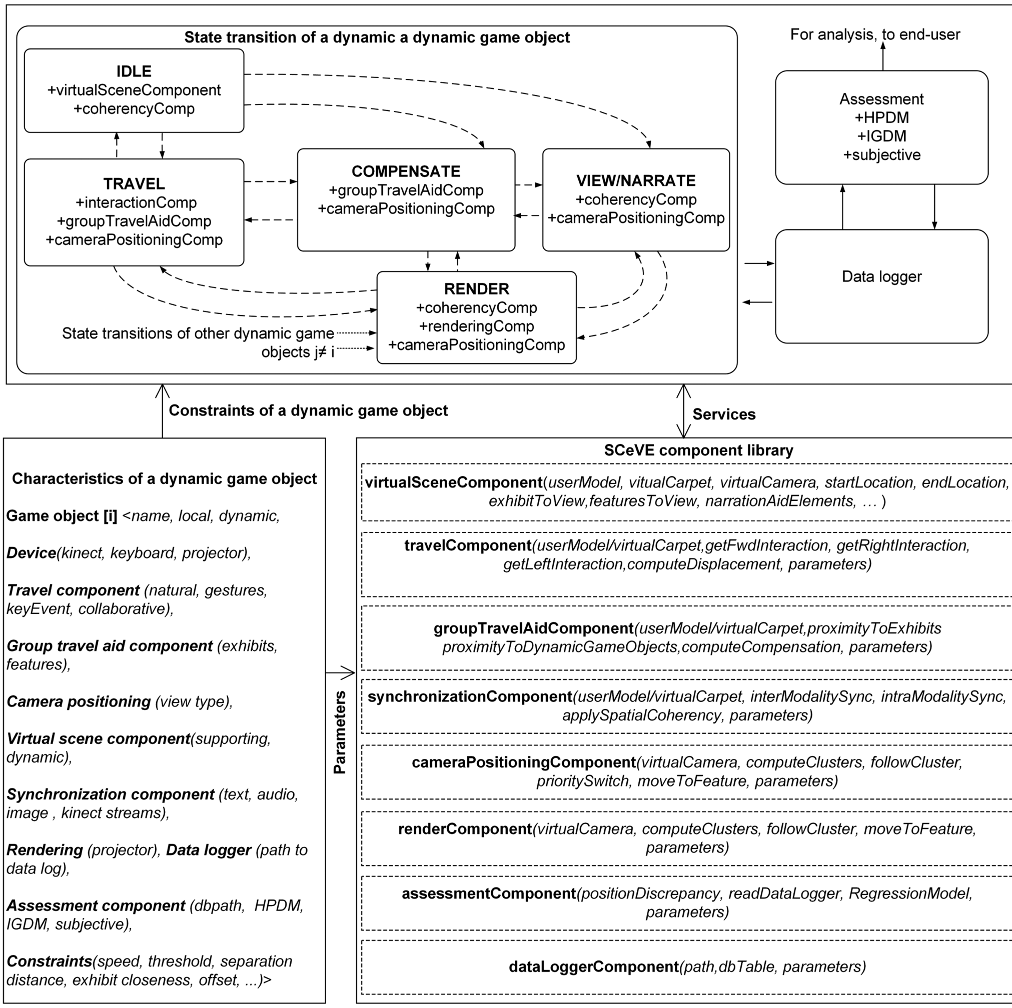

Fig. 7. The characteristics, component calls, and state transitions of a dynamic game object (a "live" model for semi-natural travel and a virtual carpet for non-natural travel) are shown.

- As game object[i] travels, services of the group travel aid component (*proximityToDynamicGameObjects*, *proximityToExhibits*) are invoked. On proximity to an exhibit or another dynamic game object, game object[i] invokes the *computeCompensation* service to temporarily slow down by reducing its travel speed to avoid missing the exhibit and to minimize passing through other dynamic game objects.
- Since the projector is the specified device for content visualization, a third-person view is selected by invoking the *followSiteUser* service of the camera positioning component.
- Anytime during the tour, the data logging service is used to log the gameobject[i] position, orientation, state transitions, closeness to the exhibits, travel path, neighbors, and so on.

Multiple instances of the plant walk tour application for various sites were created and deployed. Participants position themselves in the real-world and start to provide real-world gestures to steer their "live" model in the virtual tour. Users' explored the tour for varying network scenarios. We

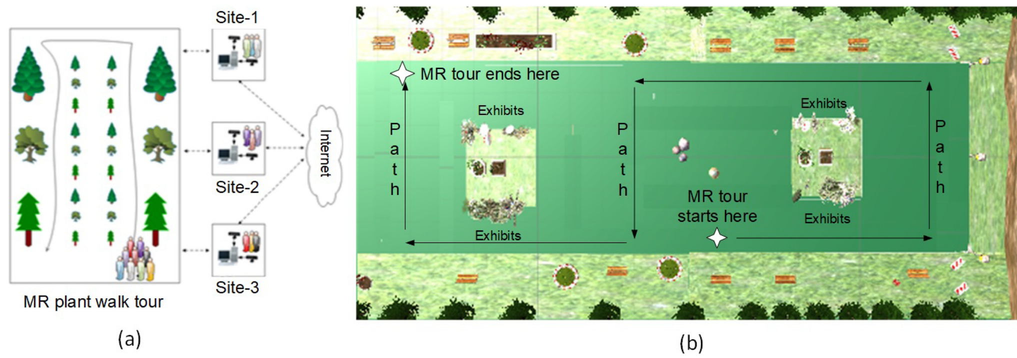

Fig. 8. (a) Shows a three-site, plant walk system setup. (b) 3D plant walk virtual environment created using Unity3D. The virtual environment consists of several supporting game objects, such as paths, tree exhibits, colliders, landmarks, panels, and other decorators to enhance look and feel of the plant walk.

calculated each user's mean speed and the interpersonal distances among groups of users using the data logs generated by the data logging service.

Case Study on the Non-natural Travel Technique. Here, we discuss how we used SCeVE to create the non-natural travel technique using the keyboard input. The design process that we followed (shown in Figure 7) is discussed below:

- Here, dynamic game object is a virtual carpet, included in the game objects hierarchy. The virtual carpet holds user models.
- We assigned the following characteristics to the virtual carpet game object: *carpet [i]<local, dynamic, device(Projector, keyboard, database), components (all), constraints(carpetspeed, threshold, separationDist, proximityDistance, exhibitCloseness, cameraOffset, refreshTime).*
- We chose a keyboard interface to provide the non-natural travel technique. Collaborating users each hold a keyboard and provide keyboard inputs to steer their carpet.
- Services *getFwdInteraction, getRightInteraction, getLeftInteraction* of the travel component move the carpet. Services such as *calculateSpatialCoherency* of the synchronization component aid in spatially synchronizing the virtual carpet with other sites.
- As the virtual carpet travels, services of the group travel aid component (*proximityToDynamicGameObjects*, *proximityToExhibits*) are invoked to calculate the carpet's proximity to an exhibit. On proximity to an exhibit, the virtual carpet invokes the *computeCompensation* service to temporarily halt the virtual carpet. This event triggers narration if the virtual carpet has invoked the *interModalitySync* service of the synchronization component.
- Since the projector is the specified content visualization device, a third-person view was selected by invoking the *computeClusters, followCluster* services of the virtual camera planning and positioning component.

4.1.2 Virtual Environment. The plant walk Virtual Environment (VE), as shown in Figure 8, was built using Unity3D. The dimensions of the VE are $100 \times 100 \times 100$ units. Exhibits are placed on either side of the pathway. The virtual environment hosts seven exhibits. Each virtual exhibit contains three distinct features to showcase. Supporting objects with the name of the exhibit is placed on exhibits. These serve as landmarks to make wayfinding easier for the user [76]. To support narration on exhibits, we include media elements such as text, image, and audio to describe each feature of all the selected exhibits. Virtual pathways and colliders guide a user's travel and prevent a user's "live" model from colliding with the other supporting objects.

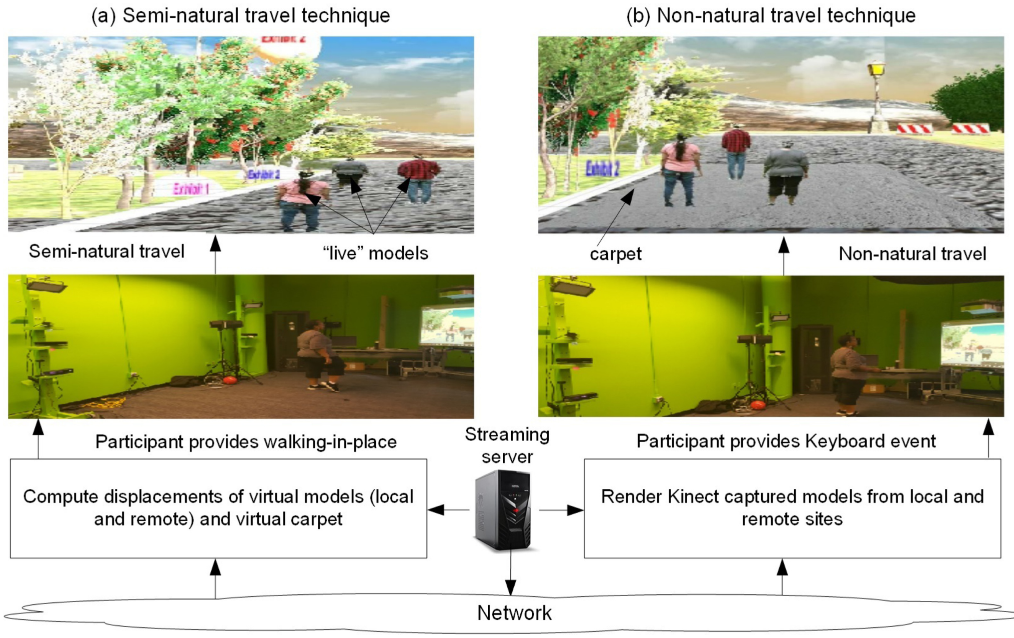

Fig. 9. A collaborative plant walk MR tour setup shown from one of the participating sites. The setup included three sites, with one user at each site. (a) Shows users taking a tour using semi-natural travel technique and (b) shows the same users taking a tour with non-natural travel technique. Kinect streams and keyboard events are communicated among the sites for travel detection and rendering, respectively.

4.1.3 *Physical Setup.* The experimental setup is shown in Figure 9. The physical setup involves three sites, with one participant at a site. At each site, a Kinect RGB-D camera is positioned behind a participant to capture the participant for the "live" avatars. The projectors were used for displaying the avatars in the virtual environment. Keyboards were provided for the non-natural travel technique. We achieved time synchronization among the three sites using NTP [48].

4.1.4 *Participants.* We recruited 18 users (6 Novices, 6 Intermediates, and 6 Experts) from the age group of 24 to 35 years to participate in the user study. There were 15 males and 3 females. Participants were computer science graduate students. All participants were right-handed and were comfortable to perform the WIP and AS gestures. Each session involved 3 users, one from each expertise level. We conducted six sessions.

4.1.5 *Procedure.* At the start of each session, the experimenter provided an overview of the gestures or the interface to use, the plant-walk setup, and the number of exhibits. After the overview, they were requested to proceed with the tour. In every session, participants were asked to take the virtual tour using both travel techniques (semi-natural and non-natural) for two network conditions (100 ms and 200 ms delay).

4.1.6 *Hypotheses.*

- **H1: The non-natural travel technique will afford the fastest personal travel speeds.** Because the non-natural travel technique requires less physical effort to employ, we hypothesized that it would allow individual users to virtually travel further in a shorter time span.
- **H2: The semi-natural travel technique will afford the closest interpersonal distances.** Because the semi-natural travel technique employs a realistic marching gesture

and users can see each other marching, we hypothesized that it would encourage groups of users to virtually travel together more than the non-natural travel technique.

- **H3: The better network condition will afford the fastest personal travel speeds.** Because of the shorter delay in feedback on the movements of all users, we hypothesized that users would hesitate less with regard to employing travel in the 100 ms condition than the 200 ms condition.
- **H4: The better network condition will afford the closest interpersonal distances.** Again, due to the shorter delay in feedback, we hypothesized that users would be able to maintain closer interpersonal distances in the 100 ms condition than the 200 ms condition.

4.1.7 Results. The individual behavior of users in a group was varied. For each metric, we conducted a two-way (technique and network delay) repeated-measures analysis of variance (ANOVA) at a 95% confidence level. Mauchly's test of sphericity indicated that the assumption of sphericity was not violated for any of the tests. Tukey's post hoc tests were used to determine which pairs of conditions were significantly different.

Individual Travel Speeds. We did not find a significant two-way interaction for technique and network delay on individual travel speeds, $F(1, 17) = 2.073$, $p = 0.168$. We did find a significant main effect for technique, $F(1, 17) = 6.818$, $p = 0.018$. The non-natural technique yielded a mean individual travel speed of 3.701 m/s^2 (SD = 1.888), while the semi-natural technique yielded a mean travel speed of 2.072 m/s^2 (SD = 1.360). However, we did not find a significant main effect for network delay, $F(1, 17) = 1.348$, $p = 0.262$. The 100 ms delay yielded a mean travel speed of 2.995 m/s^2 (SD = 1.857), while the 200 ms delay yielded a mean travel speed of 2.778 m/s^2 (SD = 1.391).

Interpersonal Distances. We found a significant two-way interaction for technique and network delay on interpersonal distances, $F(1, 17) = 5.065$, $p = 0.038$. Post hoc tests indicate that the non-natural techniques 100 ms (M = 86.299 m, SD = 50.059) and 200 ms (M = 57.141 m, SD = 29.531) conditions were significantly closer than the semi-natural technique 100 ms (M = 133.768 m, SD = 62.955) and 200 ms (M = 154.293 m, SD = 61.105) conditions.

4.1.8 Discussion.

Effect of Travel Technique. Our results indicate that the non-natural travel technique afforded significantly faster individual travel speeds and closer interpersonal distances than the semi-natural travel technique. These results support our H1 hypothesis that the non-natural technique would afford the fastest personal travel speeds due to less physical effort, but do not support our H2 hypothesis that the semi-natural technique would afford the closest interpersonal distances. We believe that less physical effort may have allowed participants to focus on interpersonal distances more with the non-natural technique than the semi-natural technique. These results also agree with prior works that indicate semi-natural techniques are inferior to non-natural techniques [47, 51].

Effect of Network Delay. Our results indicate that the network delay had no significant main effects on individual travel speeds or interpersonal distances. We did find a significant interaction effect between network delay and technique on interpersonal distance, but post hoc tests only revealed significant differences between the techniques and not the network delays within a technique. These results do not support our H3 and H4 hypotheses that the 100 ms delay would afford the fastest personal travel speeds and closest interpersonal distances due to faster feedback than the 200 ms delay. These results also do not agree with several research studies that have found latency to have a negative effect on user performance [2, 23, 25]. However, unlike prior studies, our study investigated the effects of latency on a collaborative task. Hence, we believe the natural delays induced through collaboration may have superseded the small absolute delay between our two network conditions.

SCeVE Affords Development and Evaluation. Our study was enabled by using SCeVE to rapidly develop the semi-natural and non-natural travel technique, as detailed in Section 4.1.1. Additionally, the data logging services of SCeVE enabled the real-time capture of our individual speed and interpersonal distance metrics. Analyzing these captured metrics, we were able to determine that the non-natural technique afforded significantly better individual and group performance in our study than the semi-natural technique. Hence, through these development case studies and user study, we have clearly demonstrated the capabilities of the SCeVE framework.

Scalability. Both case studies are a three-user system. The scalability of the system depends on the following: (i) physical constraints, (ii) system constraints, and (iii) network constraints. A site involving a Kinect camera can involve up to six participants. To avoid real-world collisions among participants, adequate spacing between the participants is a necessity. This adds up to the existing spacial limitation of Kinect, which is to be at least 6 feet by 6 feet. To support "live" content creation with real-time, filtering, and meshing algorithms, a sufficient GPU memory, data transfer rates between main memory, and GPU memory are mandatory. Network bandwidth in Gbps to support the collaborating sites is imperative for seamless communication and rendering among the sites. The number of Kinects and other supporting devices increases based on the application requirement.

5 RELATED WORK AND COMPARISON

Most of the existing frameworks [8, 9, 19, 26, 29, 30, 32, 40, 41, 43, 54, 55, 59, 78] follow a component-based architecture that includes software libraries and message-handling techniques to author a 2D/3D, collaborative/standalone, mobile/computer-based, first-person/third-person, HMD/projector-based, Virtual Reality (VR)/Mixed Reality (MR)/Augmented Reality (AR) custom application. As each framework is application-specific, a one-to-one comparison of the existing frameworks to SCeVE framework is not essential. We highlight the offerings similar to that of SCeVE from existing frameworks in Table 1. In our related work study, we classify the existing frameworks into the following categories:

5.1 Collaborative Frameworks that Author Education, Training, or Games

The following framework libraries author VR/MR applications that allow manipulation of a virtual object for education or online games. To mention a few, targeting 2D online games, Massively Online Games (MOG) middleware [26] offers player tracking, status update, and data management among network clients to provide a seamless first-person shooter game. Similarly, a modular architecture framework discussed in Reference [41] aids custom development of an online Chinese chess game. Focusing on a VR surface mine teleoperation, DCIVRF [9] supports (i) teleoperation, (ii) remote monitoring, and (iii) collaborative learning of surface mine equipment. The VR environment consists of a dome screen, 3D spatial sound system, two High Definition (HD) projectors, a Microsoft Surface touch table, a motion feedback chair, Inertial Measurement Unit (IMU) devices, and Global Positioning System (GPS) devices. AltSpaceVR [3] is an avatar-based, multi-user, web app offering a platform for MOG, flight simulator, learning the Chinese language, and so on.

Considering a MARS exploration application, TANDEM [29] educates children about rocks by scanning them for minerals and then displaying their composition in a bar chart graphical display. Story-telling applications such as Virtuoso (Art History Edutainment) and Enigma Rally (Museum Edutainment), Muddleware [78], a query-based edutainment framework, render animated AR agents with narration of exhibit to multiple mobile users of the same interest group. SteamVR [67] is an open-source tool kit, developed using Unity3D, enables end-users to create games. It supports hand-tracking to orient 3D objects in a virtual scene. Similarly,

Table 1. Comparison of Existing Frameworks with SCeVE Framework

Framework	Users	Visualization			Collaborative exploration		Synchronization		Evaluation	
		Virtual Scene	Camera planning	Render	Manipulation or travel	Group travel	Intra-coherency	Inter-coherency	Assess	Data logger
Frameworks that author collaboration										
4MOG	multi users	2D	first-person	PC	-	-	-	-	-	yes
Chess	multi users	2D	first-person	PC	-	-	-	chat (text-based)	-	yes
Muddleware	multi users	3D	first-person	mobile	-	-	-	text, audio	-	-
DCIVRF	multi users	3D	first-person	projector	motion feedback chair	-	-	e-book	-	-
CoRgi	multi users	3D	first-person	HMD	gesture	-	-	text, audio	menu	yes
TANDEM	multi users	2D	first-person	projector	wand	-	-	text, audio	chart	-
Frameworks that author travel										
ARToolkit	single user	3D	first-person	HMD	GPS receiver	-	-	text, images	-	-
VR Juggler	single user	CAVE	first-person	HMD	wand	-	-	text	-	-
VR JuggLua	single user	3D	first-person	desktop	sensor, devices	-	-	text	-	yes
CityViewAR	single user	2D	first-person	mobile	location sensing	-	-	images, text	-	-
SteamVR	single user	3D	first-person	HMD	teleport	-	-	text, images	-	-
VRTK	single user	3D	first-person	HMD	teleport	-	-	text, images	-	-
DWARF	single user	3D	first-person	HMD	natural	-	-	text	-	-
Colosseum3D	single user	3D	first-person	HMD	wheel chair	-	-	-	-	-
getReal3D	single user	3D	first-person	HMD	joysticks, wand	-	-	text	-	-
AltSpaceVR	multi users	3D	first-person	HMD	controllers, keyboard	-	-	text, audio	-	-
SCeVE	**multi users**	**3D**	**first-person, third-person, priority switch**	**HMD, TV, Projector**	**gesture, keyboard**	**limit speed**	**“live” , models**	**text, image, audio**	**HPDM, IGDM**	**yes**

VRTK [77] is another open-source tool kit that aids in building VR solutions including locomotion, interaction (touching, grabbing, ray-cast pointers), and controls (buttons, text) of 3D objects.

5.2 Frameworks that Author Travel

Focusing on travel, a path-planning algorithm was proposed in References [14, 43]. The algorithm provides travel assistance to visitors to either a national museum or a park. In an online portal, a user picks a spot to visit. The portal produces a tour plan and also suggests other places around the provided neighborhood. CityViewAR [40] offers web-based APIs to overlay 2D images of historic buildings on mobile clients. In an AR campus travel scenario, DWARF [8] assists a participant's travel through proprietary devices. Considering rehabilitation, research work [7] follows vehicle-based travel to improve a patient's day-to-day tasks such as sweeping the floor, setting a table for breakfast, and so on. Considering a 3D military use case, using ARToolKit [6], authors of Reference [35] provide situational awareness by overlaying real-time information on the view of the world. Reference [28] uses simple travel techniques such as joysticks and wand for avatar travel. VR Juggler [11] aids student understanding of atmospheric dynamics and visual characteristics of a tornadic storm using a wand in a CAVE immersive VR system. VR JuggLua [57] assists natural travel of 3D objects. Considering avatar-based interactions, AltSpaceVR [3] supports a non-natural travel technique.

5.3 Comparison

The collaborative frameworks that author education, training, or games aid in authoring 3D manipulation tasks of a single virtual object involving collaboration up to two users. In general, users' models lack travel, maintain a static 3D position throughout the manipulating task, and limit the rendering to a single device with a first-person visualization; whereas, the SCeVE framework facilitates multi-user collaboration involving "live" models, rendering facility extended to multiple devices, and dynamic camera viewpoint computation to distinguish travel and learning scenarios. However, SCeVE lacks manipulation of virtual exhibits; for example, rotation to provide a more favorable view.

However, the frameworks that author travel involve a single user. Involving proprietary device frameworks of this category lacks group travel involving more than two users with semi-natural and non-natural travel techniques. HMD is the preferred rendering option. As these frameworks provide only avatar-based interaction, aspects such as a sense of presence, immersion, and interactiveness with the virtual environment is a question of interest to assess. Whereas SCeVE facilitates authoring of multi-user, 3D MR tours by offering "live" model-based interactions, group travel aid render in various devices for visualization, first-person, third-person, and priority switching camera positioning and placement strategies. However, the current version of the SCeVE framework does not support wayfinding [49, 58] assistance.

6 CONCLUSION AND FUTURE WORK

In authoring collaborative MR tours, the SCeVE framework offers a suite of features to: (i) ease collaboration among the participating sites by offering synchronized interactions between sites, (ii) choose passive, semi-active, or active role of participation, (iii) support data logging and management, (iv) easy access to devices for content generation, rendering, visualization, (v) minimize development phase, and (vi) provide an organized approach to design and development. The software for the SCeVE framework is available at Reference [24]. We envision researchers can quickly build collaborative MR tours using this framework and conduct research tasks in the areas of UX (User Experience) as well as network/system performance-related experiments.

In the future, we plan to extend SCeVE offerings to similar research areas where travel is the primary means of interaction. For instance, assessing user behavior or user experience in rehabilitation, education, and gaming. Field trips enhance learning [12]. Considering (i) seasonal changes, (ii) overhead in planning, and (iii) repeated visits every semester, MR tours can provide a cost-effective solution for such virtual "field trips." Authoring an MR tour to involve seasonal changes benefits students, as they can take the tour many times to improve their subject-wise learning. Similarly, travel using real-world gestures benefits rehabilitation [15, 44, 75]. For instance, a patient's disability improvement can be modeled as an ability to navigate quickly without passing through one or more models.

REFERENCES

[1] Majed Al Zayer, Paul MacNeilage, and Eelke Folmer. 2018. Virtual locomotion: A survey. *IEEE Trans. Vis. Comput. Graph.* (2018). DOI: 10.1109/TVCG.2018.2887379

[2] Robert S. Allison, Laurence R. Harris, Michael Jenkin, Urszula Jasiobedzka, and James E. Zacher. 2001. Tolerance of temporal delay in virtual environments. In *Proceedings of the IEEE Virtual Reality Conference.* IEEE, 247–254.

[3] AltSpaceVR. 2015. AltSpaceVR. Retrieved from https://altvr.com/.

[4] Abdenour Amamra, Yacine Amara, Redha Benaissa, and Billal Merabti. 2016. Optimal camera path planning for 3D visualisation. In *Proceedings of the SAI Computing Conference (SAI'16).* IEEE, 388–393.

[5] Amazon. [2004]. Amazon Sumerian. Retrieved from https://aws.amazon.com/sumerian/.

[6] Hirokazu Kato. [2007]. Inside ARToolKit. In *Proceedings of the 1st IEEE International Workshop on Augmented Reality Toolkit.*

[7] Anders Backman. 2005. Colosseum3D-authoring framework for virtual environments. In *Proceedings of the International Workshop on Immersive Projection Technology and Eurographics Symposium on Virtual Environments (IPT/EGVE'05).* Citeseer, 225–226.

[8] Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa MacWilliams, Thomas Reicher, Stefan Riss, Christian Sandor, and Martin Wagner. 2001. Design of a component-based augmented reality framework. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality.* IEEE, 45–54.

[9] Tomasz Bednarz, Craig James, Eleonora Widzyk-Capehart, Con Caris, and Leila Alem. 2015. Distributed collaborative immersive virtual reality framework for the mining industry. In *Machine Vision and Mechatronics in Practice.* Springer, 39–48.

[10] Ian Bickerstaff. 2012. Case study: The introduction of stereoscopic games on the Sony Playstation 3. In *Stereoscopic Displays and Applications XXIII*, Vol. 8288. International Society for Optics and Photonics, 828815.

[11] Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, and Carolina Cruz-Neira. 2001. VR Juggler: A virtual platform for virtual reality application development. In *Proceedings of the IEEE Virtual Reality Conference.* IEEE, 89–96.

[12] Christoph W. Borst, Nicholas G. Lipari, and Jason W. Woodworth. 2018. Teacher-guided educational VR: Assessment of live and prerecorded teachers guiding virtual field trips. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR'18).* IEEE, 467–474.

[13] Kjell Brunnström, Mårten Sjöström, Muhammad Imran, Magnus Pettersson, and Mathias Johanson. 2018. Quality of experience for a virtual reality simulator. *Electron. Imag.* 2018, 14 (2018), 1–9.

[14] Andrew S. Cantino, David L. Roberts, and Charles L. Isbell. 2007. Autonomous nondeterministic tour guides: Improving quality of experience with TTD-MDPS. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems.* ACM, 22.

[15] Ki Hun Cho and Wan Hee Lee. 2013. Virtual walking training program using a real-world video recording for patients with chronic stroke: A pilot study. *Amer. J. Phys. Med. Rehab.* 92, 5 (2013), 371–384.

[16] Marc Christie, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, and Jonathan Pickering. 2005. Virtual camera planning: A survey. In *Proceedings of the International Symposium on Smart Graphics.* Springer, 40–52.

[17] Steven H. Collins, Peter G. Adamczyk, and Arthur D. Kuo. 2009. Dynamic arm swinging in human walking. *Proc. Roy. Soc. B: Biol. Sci.* 276, 1673 (2009), 3679–3688.

[18] Noah Coomer, Sadler Bullard, William Clinton, and Betsy Williams-Sanders. 2018. Evaluating the effects of four VR locomotion methods: Joystick, arm-cycling, point-tugging, and teleporting. In *Proceedings of the 15th ACM Symposium on Applied Perception.* ACM, 7.

[19] Nguyen-Thong Dang, Céline Chatelain, Jean-Marie Pergandi, and Daniel Mestre. 2008. A framework for design and evaluation of collaborative virtual environments. In *Proceedings of the Journées de lâ'Association Française de Réalité Virtuelle.* 119–126.

[20] Demo. [2019]. Demo. Retrieved from https://youtu.be/ie9TMDKZ750.
[21] Desiree DePriest and Karlyn Barilovits. 2011. LIVE: Xbox Kinect© virtual realities to learning games. In *Proceedings of the Theory of Cryptography Conference (TCC'11).* TCCHawaii, 48–54.
[22] Kevin Desai, Balakrishnan Prabhakaran, and Suraj Raghuraman. 2018. Skeleton-based continuous extrinsic calibration of multiple RGB-D kinect cameras. In *Proceedings of the 9th ACM Multimedia Systems Conference.* ACM, 250–257.
[23] Stephen R. Ellis, Bernard D. Adelstein, S. Baumeler, G. J. Jense, and Richard H. Jacoby. 1999. Sensor spatial distortion, visual latency, and update rate effects on 3D tracking in virtual environments. In *Proceedings of the IEEE Virtual Reality Conference.* IEEE, 218–221.
[24] Framework. 2019. Code. Retrieved from https://github.com/utd-multimedia/PlantWalk.
[25] Sebastian Friston and Anthony Steed. 2014. Measuring latency in virtual environments. *IEEE Trans. Vis. Comput. Graph.* 20, 4 (2014), 616–625.
[26] Tobias Fritsch, Carsten Magerkurth, Benjamin Voigt, and Jochen Schiller. 2007. 4MOG–massive multiplayer middleware for mobile online games. In *Proceedings of the 1st International Workshop on Intercultural Collaboration (IWIC'07).*
[27] Hania Gajewska, David P. Mendenhall, Peter A. Korn, Michael C. Albers, and Lynn Monsanto. 2003. Keyboard navigation of non-focusable components. US Patent 6,654,038.
[28] getReal3D. [n.d.]. getReal3D. Retrieved from https://projects.vrac.iastate.edu/.
[29] Benjamin Goldstein. 2000. *Tandem: A Component-based Framework for Interactive, Collaborative Virtual Reality.* Master's thesis. University of Illinois at Chicago.
[30] Michael Haller, Jürgen Zauner, Werner Hartmann, and Thomas Luckeneder. 2003. *A Generic Framework for a Training Application Based on Mixed Reality.* Technical Report. Upper Austria University of Applied Sciences.
[31] Abdelwahab Hamam and Abdulmotaleb El Saddik. 2013. Toward a mathematical model for quality of experience evaluation of haptic applications. *IEEE Trans. Instrument. Meas.* 62, 12 (2013), 3315–3322.
[32] Sylvia Irawati, Sangchul Ahn, Jinwook Kim, and Heedong Ko. 2008. Varu framework: Enabling rapid prototyping of VR, AR, and ubiquitous applications. In *Proceedings of the IEEE Virtual Reality Conference.* IEEE, 201–208.
[33] Yutaka Ishibashi, Sosuke Hoshino, Qi Zeng, Norishige Fukushima, and Shinji Sugawara. 2012. QoE assessment of fairness between players in networked game with olfaction. In *Proceedings of the 11th Workshop on Network and Systems Support for Games (NetGames'12).* IEEE, 1–2.
[34] Jeff Feasel, Mary C. Whitton, Jeremy D. Wendt. 2008. LLCM-WIP: Low-latency, continuous-motion walking-in-place. In *Proceedings of the 2008 IEEE Symposium on 3D User Interfaces.* IEEE, 97–104.
[35] Dennis Joele, G. van der Mast, and Marfa Carmen Juan-Lizandra. 2005. *Development of an Augmented Reality System Using ARToolKit and User Invisible Markers.* Research Assignment, Master Programme Media & Knowledge Engineering, Technical University of Valencia, Spain.
[36] Konstantina Kilteni, Jean-Marie Normand, Maria V. Sanchez-Vives, and Mel Slater. 2012. Extending body space in immersive virtual reality: A very long arm illusion. *PloS One* 7, 7 (2012), e40867.
[37] Yuji Kusunose, Yutaka Ishibashi, Norishige Fukushima, and Shinji Sugawara. 2010. QoE assessment in networked air hockey game with haptic media. In *Proceedings of the 9th Workshop on Network and Systems Support for Games.* IEEE, 1–2.
[38] J. LaViola. 1999. *Whole-hand and Speech Input in Virtual Environments.* Unpublished Master's Thesis, Department of Computer Science, Brown University, CS-99-15 (1999).
[39] Joseph J. LaViola Jr., Ernst Kruijff, Doug Bowman, Ivan P. Poupyrev, and Ryan P. McMahan. 2017. *3D User Interfaces: Theory and Practice (Second Edition).* Addison-Wesley.
[40] Gun A. Lee and Mark Billinghurst. 2013. A component based framework for mobile outdoor AR applications. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry.* ACM, 207–210.
[41] Wei-Po Lee, Li-Jen Liu, and Jeng-An Chiou. 2011. A component-based framework for rapidly developing online board games. *Int. J. Comput. Applic.* 33, 4 (2011), 293–302.
[42] Tsai-Yen Li and Chung-Chiang Cheng. 2008. Real-time camera planning for navigation in virtual environments. In *Proceedings of the International Symposium on Smart Graphics.* Springer, 118–129.
[43] Tsai-Yen Li . 1999. Automatically generating virtual guided tours. In *Proceedings Computer Animation.* IEEE, 99–106. DOI: 10.1109/CA.1999.781203
[44] Joan Llobera, Mar González-Franco, Daniel Perez-Marcos, Josep Valls-Solé, Mel Slater, and Maria V. Sanchez-Vives. 2013. Virtual reality for assessment of patients suffering chronic pain: A case study. *Experim. Brain Res.* 225, 1 (2013), 105–117.
[45] Gale M. Lucas, Evan Szablowski, Jonathan Gratch, Andrew Feng, Tiffany Huang, Jill Boberg, and Ari Shapiro. 2016. Do avatars that look like their users improve performance in a simulation? In *Proceedings of the International Conference on Intelligent Virtual Agents.* Springer, 351–354.
[46] Mary Lou Maher and John S. Gero. 2002. Agent models of 3D virtual worlds. CUMINCAD, IEEE. DOI: 10.1109/CW.2005.13

[47] Ryan P. McMahan, Chengyuan Lai, and Swaroop K. Pal. 2016. Interaction fidelity: The uncanny valley of virtual reality interactions. In *Proceedings of the International Conference on Virtual, Augmented and Mixed Reality*. Springer, 59–70.

[48] David L. Mills. 1991. Internet time synchronization: The network time protocol. *IEEE Trans. Commun.* 39, 10 (1991), 1482–1493.

[49] Shailey Minocha and Christopher Hardy. 2016. Navigation and wayfinding in learning spaces in 3D virtual worlds. *Learn. Virt. Worlds: Res. Applic.* (2016), 3–41. DOI: 10.15215/aupress/9781771991339.01d

[50] Alessandro Mulloni, Hartmut Seichter, and Dieter Schmalstieg. 2011. Handheld augmented reality indoor navigation with activity-based instructions. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, 211–220.

[51] Mahdi Nabiyouni, Ayshwarya Saktheeswaran, Doug A. Bowman, and Ambika Karanth. 2015. Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI'15)*. IEEE, 3–10.

[52] Niels C. Nilsson, Stefania Serafin, Morten H. Laursen, Kasper S. Pedersen, Erik Sikström, and Rolf Nordahl. 2013. Tapping-in-place: Increasing the naturalness of immersive walking-in-place locomotion through novel gestural input. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI'13)*. IEEE, 31–38.

[53] Niels Christian Nilsson, Stefania Serafin, Frank Steinicke, and Rolf Nordahl. 2018. Natural walking in virtual reality: A review. *Comput. Entert.* 16, 2 (2018), 8.

[54] Jan Ohlenburg, Iris Herbst, Irma Lindt, Thorsten Fröhlich, and Wolfgang Broll. 2004. The MORGAN framework: Enabling dynamic multi-user AR and VR projects. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, 166–169.

[55] Jong-Seung Park. 2011. AR-Room: A rapid prototyping framework for augmented reality applications. *Multim. Tools Applic.* 55, 3 (2011), 725–746.

[56] Roma Patel and Deborah Tuck. 2008. Narrating the past: Virtual environments and narrative. 249–258. http://irep.ntu.ac.uk/id/eprint/15426/.

[57] Ryan A. Pavlik and Judy M. Vance. 2012. VR JuggLua: A framework for VR applications combining Lua, openscenegraph, and VR Juggler. In *Proceedings of the 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS'12)*. IEEE, 29–35.

[58] Francesca Pazzaglia, Chiara Meneghetti, Enia Labate, and Lucia Ronconi. 2016. Are wayfinding self-efficacy, and pleasure in exploring related to shortcut finding? A study in a virtual environment. In *Spatial Cognition X*. Springer, 55–68.

[59] Wayne Piekarski and Bruce H. Thomas. 2003. An object-oriented software architecture for 3D mixed reality applications. In *Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 247–256.

[60] Suraj Raghuraman. 2017. *i3DTI: Interactive 3D Tele-Immersion*. Ph.D. Dissertation. The University of Texas at Dallas, Richardson, Texas, USA.

[61] M. J. Rorke. 2000. *Designing and Implementing a Virtual Reality Interaction Framework*. Ph.D. Dissertation. Rhodes University.

[62] Maria Roussou and Mel Slater. 2005. A virtual playground for the study of the role of interactivity in virtual learning environments. *Perspectives* 8 (2005), 9.

[63] Andrew Sears, Jinhuan Feng, Kwesi Oseitutu, and Claire-Marie Karat. 2003. Hands-free, speech-based navigation during dictation: Difficulties, consequences, and solutions. *Hum.-comput. Interact.* 18, 3 (2003), 229–257.

[64] Ari Shapiro, Andrew Feng, Ruizhe Wang, Hao Li, Mark Bolas, Gerard Medioni, and Evan Suma. 2014. Rapid avatar capture and simulation using commodity depth sensors. *Comput. Anim. Virt. Worlds* 25, 3–4 (2014), 201–211.

[65] Mya Sithu, Yutaka Ishibashi, Pingguo Huang, and Norishige Fukushima. 2015. QoE assessment of operability and fairness for soft objects in networked real-time game with haptic sense. In *Proceedings of the 21st Asia-Pacific Conference on Communications (APCC'15)*. IEEE, 570–574.

[66] Bernhard Spanlang, Xavi Navarro, Jean-Marie Normand, Sameer Kishore, Rodrigo Pizarro, and Mel Slater. 2013. Real time whole body motion mapping for avatars and robots. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*. ACM, 175–178.

[67] SteamVR. 2016. SteamVR. Retrieved from https://steamcommunity.com/steamvr.

[68] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual reality on a WIM: Interactive worlds in miniature. In *Proceedings of the International Conference on Human Factors in Computing Systems (CHI'95)*, Vol. 95. 265–272.

[69] Ayano Tatematsu, Yutaka Ishibashi, Norishige Fukushima, and Shinji Sugawara. 2010. QoE assessment in haptic media, sound, and video transmission: Influences of network latency. In *Proceedings of the IEEE International Workshop on Communications Quality and Reliability (CQR'10)*. IEEE, 1–6.

[70] Unity Technologies. 2005. Unity3D Game Engine. Retrieved from https://unity3d.com/.
[71] Unreal. 2004. Unreal game engine. Retrieved from https://www.unrealengine.com.
[72] Narasimha Raghavan Veeraragavan, Leonardo Montecchi, Nicola Nostro, Roman Vitenberg, Hein Meling, and Andrea Bondavalli. 2016. Modeling QoE in dependable tele-immersive applications: A case study of world opera. *IEEE Trans. Parallel Distrib. Syst.* 27, 9 (2016), 2667–2681.
[73] Shanthi Vellingiri and Prabhakaran Balakrishnan. 2017. Modeling user quality of experience (QoE) through position discrepancy in multi-sensorial, immersive, collaborative environments. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, 296–307.
[74] Shanthi Vellingiri and Balakrishnan Prabhakaran. 2018. Quantifying group navigation experience in collaborative augmented virtuality tours. In *Proceedings of the 3rd International Workshop on Multimedia Alternate Realities*. ACM, 3–8.
[75] Shanthi Vellingiri, Yuan Tian, and Balakrishnan Prabhakaran. 2014. A real time, distributed system with haptic interfaces for fine motor skill rehabilitation and its quality of experience. In *Proceedings of the IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE'14)*. IEEE, 53–58.
[76] Norman G. Vinson. 1999. Design guidelines for landmarks to support navigation in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 278–285.
[77] VRTK. 2016. VRTK. Retrieved from https://vrtoolkit.readme.io/.
[78] Daniel Wagner and Dieter Schmalstieg. 2007. Muddleware for prototyping mixed reality multiuser games. In *Proceedings of the IEEE Virtual Reality Conference*. IEEE, 235–238.
[79] Martin White, Emmanuel Jay, Fotis Liarokapis, Costas Kostakis, and Paul Lister. 2001. A virtual interactive teaching environment using XML and augmented reality. *Int. J. Elect. Eng. Educ.* 38, 4 (2001), 316–329.

Received July 2019; revised November 2019; accepted December 2019