Oasis: procedurally generated social virtual spaces from 3D scanned real spaces

Citation: Sra, Misha, Sergio Garrido-Jurado, and Pattie Maes, "Oasis: procedurally generated social virtual spaces from 3D scanned real spaces." IEEE transactions on visualization and computer graphics 24, 1 (2018): p. 3174-87 doi 10.1109/TVCG.2017.2762691 ©2018 Author(s)

As Published: 10.1109/TVCG.2017.2762691

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: https://hdl.handle.net/1721.1/124387

Version: Original manuscript: author's manuscript prior to formal peer review

Oasis: Procedurally Generated Social Virtual Spaces from 3D Scanned Real Spaces

Misha Sra, Sergio Garrido-Jurado and Pattie Maes

Abstract—We present Oasis, a novel system for automatically generating immersive and interactive virtual reality environments for single and multiuser experiences. Oasis enables real-walking in the generated virtual environment by capturing indoor scenes in 3D and mapping walkable areas. It makes use of available depth information for recognizing objects in the real environment which are paired with virtual counterparts to leverage the physicality of the real world, for a more immersive virtual experience. Oasis allows co-located and remotely located users to interact seamlessly and walk naturally in a shared virtual environment. Experiencing virtual reality with currently available devices can be cumbersome due to presence of objects and furniture which need to be removed every time the user wishes to use VR. Our approach is new, in that it allows casual users to easily create virtual reality environments in any indoor space without rearranging furniture or requiring specialized equipment, skill or training. We demonstrate our approach to overlay a virtual environment over an existing physical space through fully working single and multiuser systems implemented on a Tango tablet device.

Index Terms—Virtual reality, procedural generation, multiuser interaction.

1 Introduction

Following their introduction in the 1960s, head-mounted virtual reality (VR) systems have mainly focused on visual and aural senses [1], [2]. In order to enhance immersion in the virtual world, researchers have since pursued the addition of movement and haptic sense through motion platforms, exoskeletons, and other hand-held devices [3]. Realism of locomotion, a fundamental requirement for action in both real and virtual environments (VEs), had been a challenge to achieve until redirected walking, a technique that introduces a rotational gain in order to imperceptibly rotate the user away from the boundaries of the tracking space [4], was introduced. Redirected walking made possible natural and unconstrained walking in VEs without using mechanical locomotion devices. However, the technique requires a relatively large physical space and is thus not suitable for the typical home or office environment.

A key objective in VR is establishing a sense of presence. Walking is not only the most natural way of traveling, it is also a more presence-enhancing mechanism than other navigation techniques like walking-in-place and flying [5] or navigating by walk-like gestures [6].

We present Oasis, a novel pipeline to automatically generate an interactive VR experience using the physical environment as a template that allows natural and unconstrained walking in visually-immersive virtual worlds. Our approach incorporates the concept of *passive haptics* [7], i.e., receiving feedback from touching a physical object that is registered to a virtual object through object detection and tracking. In stark contrast to previous approaches built on passive haptics, where physical objects were constructed and placed in the real environment (RE) [7], our system automatically detects existing objects in the real world and places corresponding virtual objects in the VE. Users receive full haptic feedback by interacting with the real world object through its virtual proxy. Prior research suggests there are benefits to physical replication of objects in immersive virtual environments where use of actual objects significantly increased self-reported solidity and weight, not only of the object touched but also other objects in the scene [8].

While advances in consumer VR technology, e.g., HTC Vive have made it easy to accurately capture users' motions over room-sized areas using external tracking devices, the walkable area is ultimately restricted by the size of the tracked space and constrained to a fixed regular shape. We overcome this room-scale limitation by using a mobile device with inside-out tracking that allows walking to build and experience virtual worlds that can span the size of an entire house or a whole office floor.

The key contributions of our work are the following:

- A novel framework for using the physical environment as a template for automatically generating an immersive VE that conforms to any indoor space.
- An object detection and tracking pipeline for incorporating interaction, with physical objects through their virtual counterparts, in the VR experience.
- An end-to-end mobile application with the first unique combination of 3D mapping, obstacle detection, object detection and tracking, automatic VE generation, and haptic feedback.
- A multiuser implementation with combined walkable areas for shared interaction in VR.
- An asymmetrical implementation to allow VR and non-VR participants to interact in a shared virtual space.

- *M. Sra and P. Maes are with the Media Lab, Massachusetts Institute of Technology, Cambridge, MA, 02139. E-mail: {sra, pattie}@media.mit.edu*
- *S. Garrido-Jurado is with the Computing and Numerical Analysis Department from University of Córdoba, Spain. E-mail: i52gajus@uco.es*

Manuscript received June 30, 2017; revised June 30, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2762691, IEEE Transactions on Visualization and Computer Graphics

We believe our framework is the first of its kind to deliver an easy and automated mechanism for procedurally creating interactive VEs for single and multiuser experiences without any physical space shape or size limitations. Our system goes beyond the initial 3D mapping of the real world environment that detects planar surfaces (walls, floors, tabletops etc.) and adds object recognition and tracking which is missing from existing devices that also map the real world, e.g., Hololens[1], Occipital Bridge Engine[2]. With Oasis, users can create their own VR experiences, turn their living rooms into space stations, walk through the Grand Canyon with a friend or go for a stroll on Mars within minutes. We contemplate the use of our system in gaming and storytelling, education, remote tourism, architectural walkthroughs, training, and simulation. For example, space in a museum with passive haptic objects could be used to immerse visitors in any historical time period. Safety training through simulated rescue operations in replica environments is another area of interest. Generating exotic environments for watching sunsets from the comfort of a user's home could provide relaxing escapes from reality.

We demonstrate our system through the generation of four different virtual worlds based on 3D scans of physical spaces for single and multiuser experiences.

2 Related Work

The work presented in this paper attempts to simplify and automate the process of creating interactive VEs by using the real world as a template. Our system allows anyone who can use a mobile device to be able to build a VR experience compared to existing VR authoring tools that require programming and 3D modeling skills. We summarize below a few most directly related works.

2.1 3D reconstruction and object detection

The appearance of low-cost range sensors, such as the Microsoft Kinect, has provided easy access to fast and robust 3D reconstruction. KinectFusion [9] and its variants [10], [11] are among the most popular techniques for hand-held scanning of indoor scenes. More recently, a Tango[3] tablet with an integrated depth camera and inertial sensors has been used for virtual and augmented reality applications [12]. Similar to 3D reconstruction, object detection has also benefited from the availability of low-cost depth cameras. Many new approaches that use RGBD data from depth sensors have been proposed for object detection [13], [14], and they generally provide more robust detection than possible with 2D images. The Sliding Shapes detector [15] extends the 2D sliding window approach for object detection in images to depth maps. A window is moved along the 3D space in a point cloud and evaluated by a classifier at each position. An ensemble of exemplar SVMs (Support Vector Machines) is used to decide if the window contains an object.

2.2 Real walking in VR

Natural walking is a desired feature in many VR applications [5] but remains a challenge because of space and tracking requirements. Redirected walking makes natural walking in VEs possible by tracking and manipulating the user's real world trajectory [4]. The recently introduced consumer HTC Vive[4] system allows a user to move in a small tracked space with a maximum size of 5×5 m. For small spaces, low cost depth sensors have been used to track users [16]. The idea of applying 3D analysis to determine walkable area is outlined by Nescher [17] although its implementation is not presented. Change blindness, a perceptual phenomenon that occurs when a person fails to detect a visual change to an object or scene was used to allow a user to walk through a virtual environment that is an order of magnitude larger than the physical space [18].

2.3 Passive Haptics in VR

Passive haptics have been shown to both enhance immersion in VR and also make virtual tasks easier to accomplish by providing haptic feedback [19]. Adding representations of real objects, that can be touched, to immersive VE enhanced the feeling of presence in those environments [8]. Low-resolution physical models made of styrofoam and plywood were found to significantly improve presence [7]. TurkDeck used ”human actuators” to operate physical props in real-time [20]. Substitutional Reality pairs every physical object surrounding a user to a virtual counterpart [21]. In Annexing Reality [22], the system detects simple geometry primitives (e.g., cylinders, cones) from objects placed on a table to match known virtual objects.

2.4 Adaptive Systems

HTC Vive is a recently introduced consumer VR device that allows developers to create experiences using natural locomotion in a room sized space. The Vive Lighthouse tracking system requires users to manually trace out a play area, clear of furniture and obstacles, using the included hand-held devices. If the area is oddly shaped, the largest square or rectangle (maximum size 15'x15') that can fit in the space is chosen and the VR app is loaded accordingly. Hololens and Occipital Bridge Engine, both augmented reality (AR) devices, analyze the user's physical environment to find flat surfaces like a wall or a couch seat to determine potential locations for placing virtual objects in the real world, as described in FLARE [23]. SnapToReality [24] presents a related AR system where edges and planes in the physical environment are detected for aligning virtual content placement.

We designed our system to overcome the limitations of existing VR experiences that incorporate passive haptics or natural locomotion. Unlike the Vive, the Oasis tracked physical area can be regular or irregularly shaped, e.g., a square or an octagon with a hole with no limitation on size. The user does not need to prepare a play area in their living room by rearranging furniture as our system uses any given physical space as input and automatically layers a corresponding virtual world above it. While the

[1] Hololens. https://www.microsoft.com/microsoft-hololens/en-us
[2] Occipital. http://structure.io/developers#bridge
[3] Tango. https://developers.google.com/tango/
[4] HTC Vive. https://www.htcvive.com/

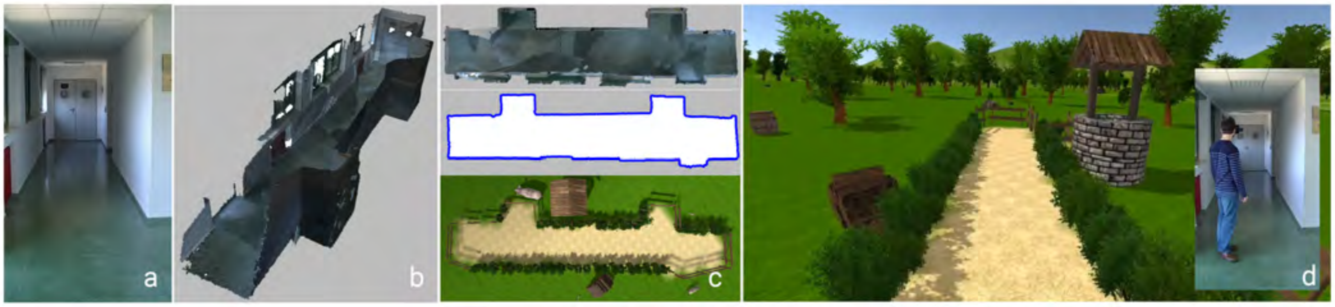

Fig. 1. Steps of the proposed system shown in order of progression from left to right. (a-b) We start with creating a 3D map of the real environment. (c) We detect the walkable area in the input 3D map to determine where the user can move freely. The generated virtual world is created according to the estimated walkable area in the point cloud. (d) Inset shows a user navigating the generated virtual environment by walking in the real environment, while visually experiencing it through a Tango HMD.

Vive is tethered to a PC and the user needs to stay within the Lighthouse tracked volume, users in our system are free to move from room to room to generate large virtual worlds. This is possible because of the Tango inside-out tracking and does not need an external tracking system.

Even though we analyze the physical environment for planar surfaces, similar to the Hololens or Occipital, we use the output as a template to create a complete virtual environment, with visual indications of where a user can and cannot go. Since the VR user cannot see the physical world, we must build our virtual world to provide safe travel for all users, whether in the same physical space or remotely located. We go further, and add interaction with automatically detected real world objects through their virtual counterparts, something neither of the above mentioned devices does. While current object recognition systems can achieve remarkable recognition performance for individual classes, e.g., chairs in our case, the simultaneous recognition of multiple classes remains a major challenge. We focus on recognizing only one class of objects as our primary goal is to demonstrate user interaction with daily objects in their natural environment through passive haptics either for single or multiuser scenarios.

Our target space is home or work environments where it is impractical to radically rearrange furniture, build matching props, or use ”human actuators”. We have not found any approach like ours in the literature that automatically generates a VE using physical space as input with walkable area segmentation, object detection, and object tracking for passive haptics.

2.5 Social VR and Asymmetrical VR

An early form of social VR was the text-based MUDs or Multi-user Dungeons where many users shared the computing environment. While current social VR environments like Facebook Spaces[5] or AltspaceVR[6] are graphical and users can create and customize their avatars, the underlying concept of interacting with remotely located users in a shared virtual space is the same. Diamond Park was a social VR system in which geographically separated users could speak to each other and participate in joint activities like cycling [25]. Newer systems like Metaspace I and Metaspace II allow co-located users with full-body avatars to interact in a room-scale VE with each other and with objects in the room [16]. In a simulated snowball fight, two users play the roles of ‘shooter’ and ‘target’ where roles are chosen based on the size of each user’s RE [26]. This allows someone with standing room only to play with someone who has a larger tracked space in VR. The Oasis multiuser version allows co-located and remotely located users to collaborate and interact in a VE while also allowing non-VR users to participate in an asymmetrical shared virtual experience.

3 System Overview

In this section we describe our virtual reality generation system, designed for creating interactive VEs that allow users to walk and interact with objects in the real world while being visually immersed in the virtual world. The main idea is to use the physical world as a template for the VE so as to create a correspondence in scale, spatial layout, and object placement between the two spaces. Figure 1d shows a user immersed in one of the generated experiences. The user can freely walk, sit, bend down, or turn and tilt their head. A Tango device continuously tracks the user’s position and provides them with a first-person view into the VE.

One main aspect of our system is that it provides the user a physical/haptic experience along with a visual and auditory one. Our system achieves this through real-walking in the VE and *passive haptics*, i.e., whenever users touch a particular object in the virtual world, they also touch a corresponding object in the real world. Figure 13b illustrates a user feeling the realness of a virtual chair in our generated VE by touching the corresponding real chair and sitting down.

The system automatically detects walkable areas (WAs) of any shape and size from 3D scan data allowing users to walk freely using inside-out tracking in the WAs. This is unlike existing approaches for walking to navigate in a VE that require a tracking system tied to a fixed size tracked space. Our use of object detection for passive haptics is also different from previous systems where the RE is constructed with low resolution physical props to match the design of the VE [7].

[5] https://www.facebook.com/spaces
[6] https://altvr.com

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2762691, IEEE Transactions on Visualization and Computer Graphics

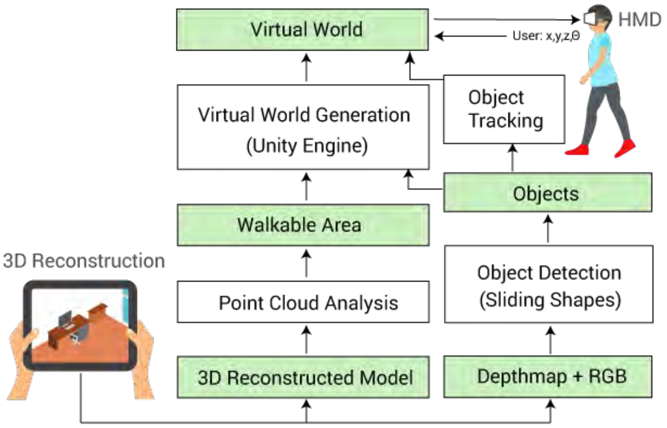

Fig. 2. Full process diagram starting with a 3D scan of the real environment (bottom left) to the user walking with an HMD in the generated virtual environment (top right). Green boxes represent data while white ones represent sub processes in our system.

Oasis allows users from different physical environments to connect, interact, and play in a shared virtual world that has been adapted to each user's space. We consider three scenarios for interaction between co-located and remotely located users: 1) where several users share the same RE, 2) where the users share the same VE but they are located in different REs, and 3) an asymmetrical case where non-VR users participate in the VR experience through a standard PC or tablet device. Oasis automatically handles the different REs and creates a single virtual world that is shared by all users where they can interact in real time.

The software pipeline (Fig. 2) includes: (i) building a 3D map of all the REs using a Tango device, (ii) analyzing the 3D data to determine WAs, i.e., spaces that are free of obstacles like walls or furniture, (iii) creating combined WAs for multiuser scenarios, (iv) using the depth data from the RE scans to do object detection and tracking, (v) using the mapped WAs to procedurally generate a VE, (vi) placing virtual models of detected objects in the VE, and (vii) tracking the users and the objects in real-time as they interact with the VE and with one another through the Tango placed in a viewer to function as a head-mounted display (HMD). In the multiuser cases, the position of each user in the VE is shared over the network with all users and communication between users happens over Discord. [7]

Oasis allows three types of interactions with the procedurally generated VE, namely (i) interaction with digitally created elements, (ii) interaction with elements that have real world counterparts, and (iii) interaction between users. Digitally created elements, e.g., wildlife, respond to the user's presence. Thus, interaction with them is based on the user's location in the VE as well as the user's proximity to the virtual elements. For objects that have real world counterparts, e.g., the chair, once the user puts on the HMD, the physical objects are never seen directly, only felt. Interaction with them happens through their virtual representations. Finally, interaction between users is implemented based on their relative position and orientation in the VE.

4 Technical Details

In this section, we describe the key components of our system.

4.1 Walkable Area (WA) Detection

The main input for our system is a 3D reconstruction of the physical environment. For acquiring input data, we use the Tango device which features a motion and depth sensing camera to create a 3D map of the environment. The integrated sensors continuously return the 3D position and orientation of the device producing a registered point cloud of the environment in real-time. A point cloud created through any other 3D scanning technique or device would also be valid input for our system.

We detect the walkable area in the reconstructed 3D model and use it to procedurally generate a virtual world. For remotely located users with dissimilar physical spaces, a walkable area is determined individually for each RE and then combined as detailed in Section 4.2.

Figure 3 illustrates how we determine the walkable area, the space that is free of obstacles like furniture, in the 3D map of a living room. We focus on detecting open space and use boundary elements (Section 4.3.2) to create a visual barrier between the detected open space and the obstacles like furniture or walls. We begin by pre-processing the input point cloud, $\mathcal{P} = \{\mathbf{x}_i \in \mathbb{R}^3\}$, which involves removing isolated components with bounding diameters smaller than an empirically determined minimum size of 2 cm. Using the pre-processed point cloud, we separate the floor from the rest of the elements in the RE, i.e., furniture, walls and other obstacles. The floor isolation is achieved by fitting a plane $\mathcal{Q}$ to the points in $\mathcal{P}$ using RANSAC sample consensus.

The floor point cloud, $\mathcal{P}_{floor}$, is composed of those points in $\mathcal{P}$ with a distance to plane $\mathcal{Q}$ shorter than a threshold ε, i.e., $\mathcal{P}_{floor} = \{\mathbf{x}_i \in \mathcal{P} \mid \mathfrak{D}(\mathcal{Q}, \mathbf{x}_i) < \varepsilon\}$, where $\mathfrak{D}$ is the orthogonal distance function between a plane and a point. A minimum distance of $\varepsilon = 0.05$ m was employed in the processing of the 3D reconstructed models of our indoor scenes.

The rest of the points in $\mathcal{P}$ belong to potential obstacles. Objects that are above the user's head, like the ceiling, do not impact the WA and are ignored since there is no possibility of collision with them. We define the point cloud of obstacles as $\mathcal{P}_{obst} = \{\mathbf{x}_i \in (\mathcal{P} - \mathcal{P}_{floor}) \mid \mathfrak{D}(\mathcal{Q}, \mathbf{x}_i) < h\}$, where h is the minimum height. The value of h can be incremented if we want to consider the possibility of a user jumping in the VE. Figures 3b and 3c show $\mathcal{P}_{floor}$ and $\mathcal{P}_{obst}$ respectively.

A top view of the WA provides a simplified representation while maintaining the important information needed to replicate the space in the generated VE. To get the top view image, I_k, of a point cloud $\mathcal{P}_k$, we project the 3D points onto the floor plane $\mathcal{Q}$.

Figures 3(d-e) show the top binary images for the floor and obstacle point clouds, i.e., I_{floor} and I_{obst}.

By combining these two binary images we obtain a top view of the WA in the RE. We combine the images as $I_{WA} = I_{floor}(1 - I_{obst})$.

I_{WA} represents the areas in the 3D reconstructed model where a floor has been detected and there are no obstacles

[7] https://discordapp.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2762691, IEEE Transactions on Visualization and Computer Graphics

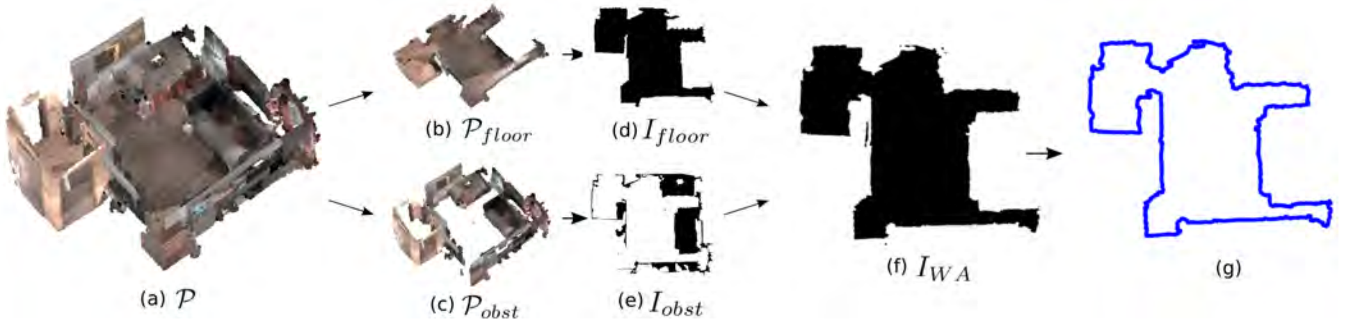

Fig. 3. Walkable area detection. (a) 3D reconstructed model. (b-c) Floor and obstacle point clouds. (d-e) Top views of the floor and obstacles. (f) Segmented walkable area using d and e. (g) Contour of detected walkable area.

like walls or furniture (Fig. 3f). The rest of the space outside this area is occupied by an obstacle or its status is unknown. We use this binary image as a guide for generating the VE. To account for errors in reconstruction and tracking, we apply an erode filter to shrink the WA. This increases the distance between the WA and the obstacles in the environment for a safer immersive experience. To segment the exact boundaries of the WA for the VE generation process, we detect contours in I_{WA} (Fig 3g) and only keep the one which outlines the largest WA in the environment. When present, we keep internal contours representing holes in the WA, as they can lead to the generation of interesting VEs such as inland lakes or craters.

4.2 Combined Walkable Area Estimation

One of the available multiuser modes in Oasis consists of two or more users sharing the same VE but different REs. We combine the different WAs of each RE into a single joint WA that is then used by all participants, as shown in Figure 4. This is useful for scenarios where users need to cooperatively do a task in VR which requires them to be in close proximity. Unlike current social VR experiences where teleportation is a common navigation mechanic, our goal is to allow all participants to use real walking to navigate the VE. While real walking can lead to a more natural shared VR experience, there are implementation challenges related to creating a combined WA due to the differences in the size and shape of each user's RE.

The simplest solution is combining the WAs by arranging them near one another without intersection as shown in Figure 4c and d. This way, users can see each other in the VE from their own WA but they cannot interact closely because there is no overlapping physical space that maps to shared virtual terrain. This solution limits the interaction possibilities between users as they cannot walk over to each other's space. However, it is convenient in cases where direct user interaction is not essential and conserving the maximum size of each WA is important.

A more challenging case involves merging the different WAs into a single WA such that the common physical space is maximized for each user. In other words, the combined WA is the intersection of all the individual WAs with the largest area as shown in Figure 4g. This means that most users will not be able to use the full extent of their individual WA in favor of maximizing the global shared space for all the participants. This is necessary when users need to interact with one another and the interactions require proximity in the VE.

To find the maximum intersection between two WAs, all possible 2D translations and rotations of one WA relative to the other are considered. In practice, the number of combinations can be huge making the search of the largest intersection considerably slow. Consequently, we first perform a coarse search using low resolution WAs and higher increments in translation and rotation. Concretely, we consider translation increments of $5cm$ and rotation increments of 1 degree. Once we find an optimal coarse intersection, we refine it using the original higher resolution WAs with smaller increments in translation and rotation around the coarse solution.

Oasis generates a multiuser VE from the combined WA and shares it with all the users. To create a combined WA, all individual WAs must be available beforehand. This process takes only a few seconds though the time taken can increase as the number of users increases. When there are more than two REs, the search space grows exponentially. It is thus preferable to add a new WA incrementally by finding its intersection with the last combined WA to minimize the time taken for creating the final combined WA.

Finally, there are some special cases that need to be considered. The intersection of two WAs can generate a new WA with two (or more) spaces that are not connected. For user interaction, only one of these spaces is employed, so only the largest one is considered to estimate the largest intersection.

Furthermore, it can happen that the two spaces are connected by a narrow stripe that cannot be crossed by a person without colliding with obstacles outside the WA. This could lead to miscalculating the maximum area and, to avoid it, we estimate the area after shrinking the WA intersection by 30cm, which is enough space to allow a person to pass through.

4.3 Virtual World Generation

After analyzing the 3D reconstructed model we generate the VE as described below.

We use the estimated WA as the central element for our procedural map generation. The visual aesthetic of the virtual world is determined by the design of the available set of models, textures, etc.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2762691, IEEE Transactions on Visualization and Computer Graphics

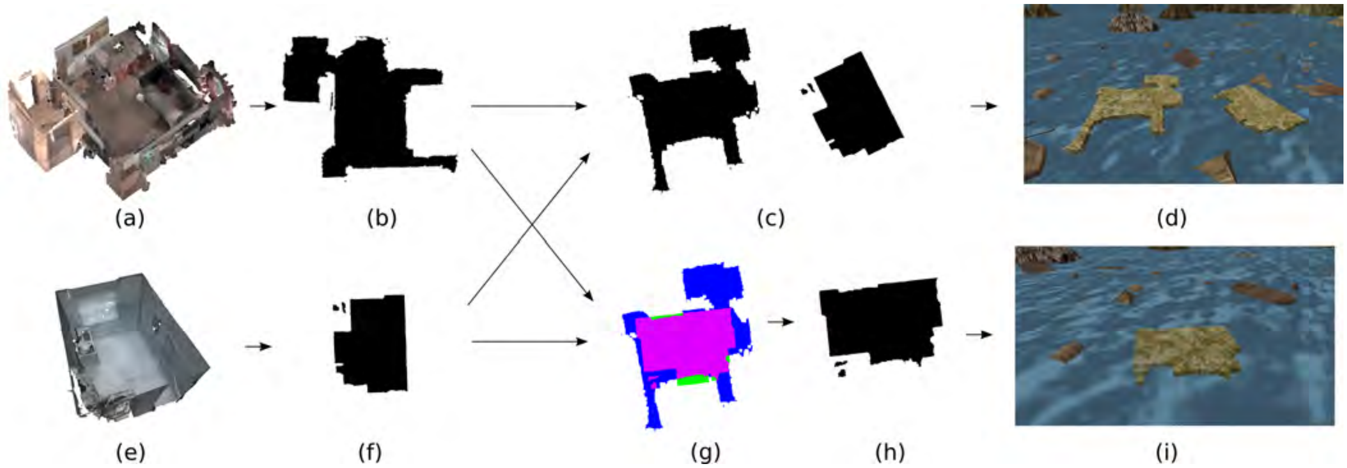

Fig. 4. Combining two WAs to create a single shared WA. (a,e) 3D reconstructions of two different REs. (b,f) Estimated WAs for each RE. (c,d) VE generated for the simplest solution of WA combination, where the two WAs are arranged next to each other without intersection. (g) Maximum intersection of the two WAs. In blue and green the WAs of (a) and (e) respectively and in pink their intersection. (h) Combined WA based on the maximum intersection, equivalent to pink area in (g). (i) VE generated using the combined WA in (h).

The generated VE is composed of three types of elements: (i) static elements, (ii) WA boundary elements, and (iii) other virtual world elements. Each type of element serves a specific purpose. It is either visual, interactive, functional or a combination thereof and impacts the user's immersive experience as described in each of the following sections.

4.3.1 Static elements

Static elements are visual elements that stay at the same position, orientation, and scale following each generation of the VE. Some examples of static elements in our generated VE are the skybox, mountains in the distance, and the ground terrain (see Figure 5a). In our design, these elements are usually visible to the user from a distance and influence only the visual feel of the virtual space.

4.3.2 Boundary elements

An important feature of our generated VE is the capacity of the user to move freely within the mapped WA without fear of colliding with any object or wall in the real world. This feature calls for design techniques that prevent the user from walking into occupied areas while they are visually-immersed in the VE.

We use virtual elements, referred to as boundary elements, that indicate areas in the virtual world the user should not access. Boundary elements are both visual and functional and, hence, the type (e.g., fence, shrub, water) of virtual items used is important. Since people do not usually walk through obstacles in the real world, we use common sense knowledge of real world behaviors as well as simple design techniques from video games to place appropriate boundary objects in generated VEs.

The output VE can be an outdoor environment (e.g., a forest or a Martian landscape) or an indoor one (e.g., a cave or a space station), independent of the type of physical environment of the user. We focus on creating virtual worlds that look and feel much bigger than the 3D mapped indoor spaces they are based on.

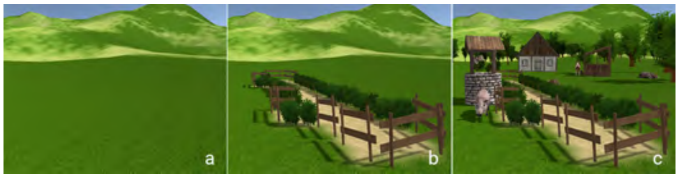

Fig. 5. Virtual world generation for environment in Figure 1. (a) Static elements like terrain and mountains are placed first. (b) Boundary elements like fences, and shrubs are positioned next. (c) Finally, rest of the world elements are filled in to create a VE that feels alive.

A concern in designing visually expansive spaces is that even though the users are aware that they should not pass through the boundary barriers, there is some chance that they may touch or lean on those barriers, resulting in them touching furniture or walls in the real world. The disconnect between touching, for example, a fence and touching a piece of furniture in the real world would be disruptive for immersion. A potential solution would be to use passive haptics to replace all obstacles in the real world with corresponding virtual counterparts, e.g., virtual walls for physical walls similar. This, however, would constrain the generated VEs to indoor environments. To reconcile generating large open spaces while still providing tactile feedback, we employ passive haptics for the interactive objects in the scene, e.g., a chair in this demo.

In the farm VE in Figure 5b, a wooden fence and shrubs are placed around the WA to prevent users from leaving that space. The elements visually and functionally blend in with the design of the VE and do not command any particular attention. Since, fences and shrubs are commonly used in the real world for enclosing a space, we expect the user to behave in a manner similar to their real world behavior when they encounter them in our VE. We believe, this can help prevent some, if not all, potential mishaps by keeping users within the WA.

The design and type of these boundary elements varies based on the design of the virtual world. For example, the water in Figure 6b or lava in Figure 6c acts as a virtual

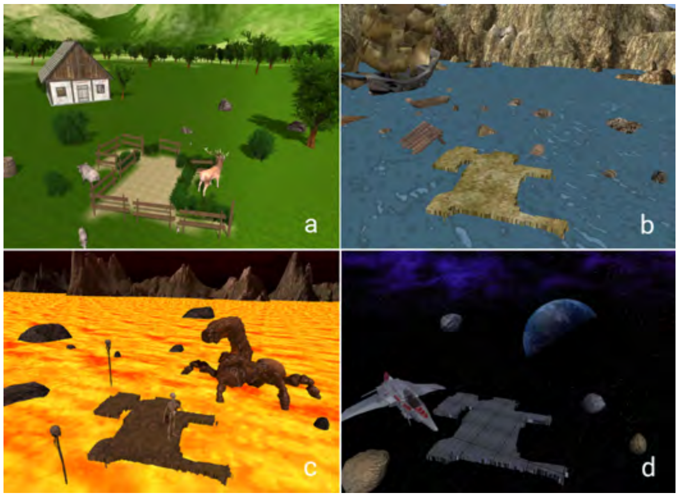

Fig. 6. Four different virtual worlds generated for the same real environment (Figure 3a) using different visual styles and procedural generation rules. Each world has a different set of boundary elements that are contextually appropriate, (a) fences and shrubs, (b) water, (c) lava, and (d) dark space.

boundary and prevents users from walking into walls or furniture in the real world. Other boundary element examples are the blackness of space, a canyon ledge, red rope barriers with post stanchions in a virtual museum or rocks in a virtual cave. In addition to boundary elements, we also differentiate between the floor texture of the WA from the rest of the open space in the VE to reinforce the spatial demarcation.

Before automatically placing the boundary elements in the scene, we simplify the contour of the WA by performing a polygonal approximation. The type of boundary element to be placed is selected using contour parameters like angle or segment length. For example, in Fig 5, fences look natural when placed along straight edges but intersect awkwardly when placed at contour corners with acute angles. We therefore include softer elements like shrubs for placement in the corners. They are smaller, fit in the tight space and even if two shrubs intersect, they do not look unnatural. For some VEs, polygonal approximation is a necessary step needed to produce a more pleasant and natural looking visualization. In other VEs, we may not need any boundary elements and polygonal approximation. For example, in the floating island VE (Fig. 6b) we use water as a boundary element and the noisy contours do not effect the visual outcome negatively. In fact, the jaggedness adds to the visual realism of the floating island.

4.3.3 Virtual world elements

Once the static and boundary elements are in place, we add the rest of the elements in the environment as shown in Figure 5c. These elements impact the user's visual and interactive experience in the VE. Placement of the world elements is a design task. VE generation consists of an optimization process to obtain the virtual world that best accomplishes defined rules.

Techniques for procedural map generation usually employ a set of rules that describe the desired features in the output map. We create a set of rules that define spatial relationships between sets of virtual elements. The rules also take into account proximity of items to the WA and their orientation relative to the WA. For example, in the farm VE, the door of the house always faces the fenced area. We created 25 different rules for the placement of elements in the farm virtual world shown in Figure 5c. Some examples of the rules employed are

```
1. Virtual elements do not intersect with the WA.
2. Animals are placed close to the WA.
3. House faces the WA.
4. No trees between the house and the WA
   (so that visibility is not impacted).
5. Baby boars close to the mother boar.
6. Pail close to the well.
7. Horse close to the shelter
   etc.
```

To generate an aesthetically pleasing virtual world map, we need to optimize the position of the elements based on the designed rules. We chose to use a genetic algorithm (GA) [27] because it allowed us to model the optimization function more explicitly than other optimization approaches. We also wanted to use a more sophisticated method than a simple heuristic/random approach. A near optimal solution provided by the GA was preferred because it created a different VE after each execution.

We employed a GA with elitism, i.e., the best individual remains in the next generation, and a maximum number of iterations as stop criteria. Each individual is composed of the poses of N elements, i.e., all the virtual elements in the scene. Each pose is a 2D transformation matrix describing the translation, position, and scale of each virtual element as viewed from the top. The values for the initial population are generated randomly.

The crossover operator is performed by two individuals from the previous generation selected by fitness ranking. A uniform crossover approach [28] is performed so that the N virtual elements of each parent are randomly distributed between the two new children. The mutation is randomly applied to each new individual, producing a small modification in position, rotation or scale. Finally, the fitness function evaluates the optimality of an individual (a virtual world composed of N virtual elements) with respect to the set of rules. For each rule r_i we define a function $f_i \in [0, 1]$ that provides a score for that rule based on the current individual. For example, a function to evaluate a minimum distance between two elements (e.g., rule 6) is expressed as a ramp function. The fitness function for an individual is then defined as the weighted average of each of the rule functions f_i, that assign a different weight, w_i, to each rule depending on its importance: $\sum_{i=0}^{N} w_i * f_i$

We use the best individual from the last generation to place the virtual elements in the generated VE. Figure 6a shows the virtual world generated for the living room environment in Figure 3a. By altering the style (meshes, textures, shaders) and the rules we can generate totally different worlds for the same RE (Figs. 6b-c-d).

4.4 Collision Prevention

To prevent collisions with real world obstacles, the HTC Vive includes a ”chaperone” system. It works by displaying a wall-like blue grid in the user's virtual vision when they

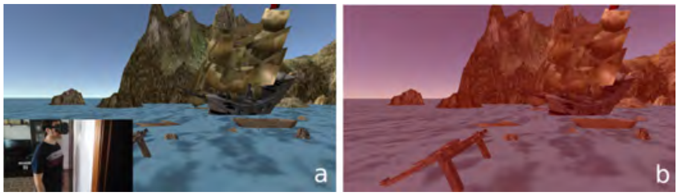

Fig. 7. The visual warning system. (a) User's view when approaching a wall with the warning system not yet enabled. (b) User's view when the warning system is enabled. A red tone covers the camera view to warn the user.

are in close proximity to the boundaries of their configured play area. If the user gets closer still to the boundary, the forward-facing camera gives them a sort of thermal view of their surroundings. We did not find any studies that explore if and how the "chaperone" system affects immersion in VR. However, in our experience some users chose to disable it for greater immersion, even at the risk of potential collisions with real world obstacles. In addition to overstepping the tracking area bounds, there is the potential obstacle of having a cord trailing the user around the room, which the Tango does not have, though wireless solutions for desktop VR devices have recently been announced.

Although boundary elements may be enough to avoid collisions in most cases, Oasis also implements a gradual warning system composed of three phases namely visual, aural, and haptic. Each phase is activated progressively from the least invasive to the most, as the user gets closer to a physical obstacle. In the first phase, the camera view is covered by a red filter when the user approaches an obstacle. The intensity of red increases as the user gets closer to the boundary (Fig. 7). In the second phase, an audio warning is emitted and the volume increases as the user gets closer to the obstacle. In the last phase, when a collision is imminent, a haptic signal is sent through vibration of the Tango tablet. Despite the warnings, it is possible that the user will reach out and touch an obstacle or their knee will inadvertently touch an obstacle especially when they are represented by water or other similar ground level boundary elements. In these situations, touching the obstacle will provide haptic feedback and deter further movement in that direction. However, we believe the previous warning phases will have been alarming enough to slow down the user's movements to prevent any dangerous collisions.

4.5 Object Detection and Tracking

Object detection in the real world for use in our VE is a challenging task due to variations of viewpoint, occlusion, self-occlusion and sensor noise to name a few. We use the Sliding Shapes object detector [15]. This method slides a window along the 3D space of a depth map. Each position is evaluated using an ensemble of SVM classifiers to determine if the window contains a trained object. The positions with a score over a threshold are considered and non-maximum suppression is applied to remove duplicate detections. Figure 8 shows the input and output of the detection process.

Similar to many classification techniques, Sliding Shapes requires an initial training step. It uses a large set of synthetically generated object models of typical indoor objects (e.g., chairs, tables, etc). Hence, objects that are considered for recognition and thereby interaction in our VE need to be selected a priori in order to perform offline training and classification. An automated system could ideally detect which objects are available in the room, select some of them for inclusion in the design, and choose object models which would be appropriate candidates as their virtual counterparts, using a rule-based system.

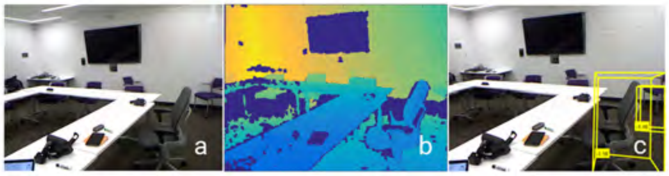

Fig. 8. Example of chair detection using the Sliding Shapes detector (a) RGB image. (b) Depth image (c) Detected chairs.

We capture several depth maps of the user's environment during the 3D scanning process for classifying objects in the scene. The final output after classification is the position, orientation and scale of a detected object that we use to position a similar object model in our VE at the beginning. The point cloud is also used for object tracking when the user is in VR mode.

In our example output, we detect a chair in the real world and place a corresponding chair in the VE at the same scale, position and orientation. We compare the object in the scene with 880 instances of chairs during classification to recognize not only that a particular object is a chair but also what type of chair, e.g., an office chair with wheels and armrests. This allows us to better match a virtual counterpart to the detected object and also enables detection of a very wide range of chairs. Detected objects are tracked in real-time when the user is in VR mode. We store the point cloud of the detected object in its original position and look for the object in each subsequent depth frame. Real-time tracking is achieved using ICP (Iterative Closest Point) [29] which provides the relative 3D transformation between two poses from two point clouds. We compare the depth map provided by Tango with the stored point cloud of the object to be tracked (e.g., chair). If a change in pose is detected, we update the virtual object's position accordingly. To improve performance and robustness, we only perform ICP when the user is looking at the object and we crop the input depth map to the volume surrounding object at its current position. ICP is unable to converge for large or fast object movements, especially since it is running on a mobile device, unlike the GPU based implementations like KinectFusion [9]. While we have built marker-based tracking [30] to overcome this limitation, the need for adding markers in the user's environment takes away from the goal of creating a fully automated system for VR world generation.

5 User Experience

An important feature of our system is its capability to support user interaction with physical elements, virtual elements, and other participants. We describe three interaction scenarios to highlight the possibilities of the VEs generated using our approach.

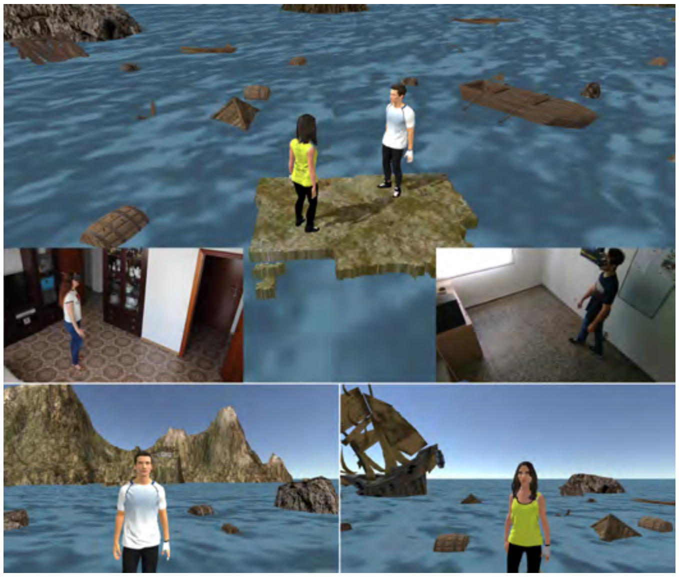

Fig. 9. Two users (Fig. 4i) share the VE from different REs. Top image and insets show the VE with the users' avatars and the position of the users in their REs. Bottom images show the first person point of view of each user.

Upon start of the application, the users enter a virtual world from a fixed position in the physical world that was mapped earlier and observes the generated outdoors through the Tango HMD. The objective behind adding passive haptics, i.e., when users touch or manipulate an object in the virtual world they simultaneously touch or manipulate a corresponding object in the physical world, is to embrace the domestic environment filled with furniture and objects by making it part of the users' experience.

In a generated VE, the detected object appears as a 3D model of the real one. It thus, presents the same affordances and allows the same interaction as its physical counterpart. The concept of *affordance* introduced by Gibson [31] refers to the interaction possibilities perceived by the observer of an object. For instance, a chair affords sitting.

5.1 Multiuser Interaction

A significant element in Oasis is the implementation of a multiuser system which allows interaction between different users in the same VR world. Among the many possibilities, we focus on the following three:

1) All users are located in different physical spaces but share the same VE, as shown in Figures 9-10. In this case, the VE is automatically adapted to the different REs through a combined WA as explained in Section 4.2.
2) Two or more users share the same physical space as shown in Figure 11. In this case it is not required to adapt the REs as there is only one physical space. Users walk in the same space and interact through voice and touch as they would in real life.
3) One user is connected through VR while another uses a different modality to access the virtual world like a PC or a tablet in an asymmetrical VR scenario, as shown in Figure 12.

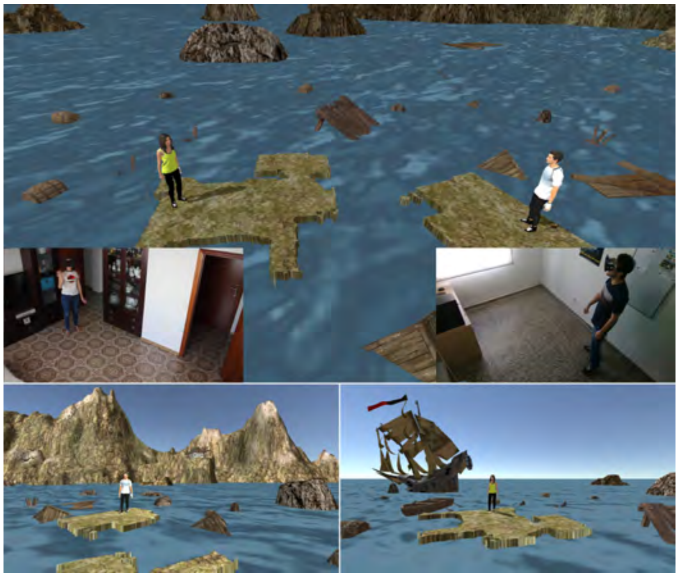

Fig. 10. Shared VE (Fig. 4d) where the two WAs are not combined but arranged separately. As in Figure 9, two users are in different REs. Top images show the VE with the users' avatars and the position of the users in their REs. Bottom images show the first person point of view of each user.

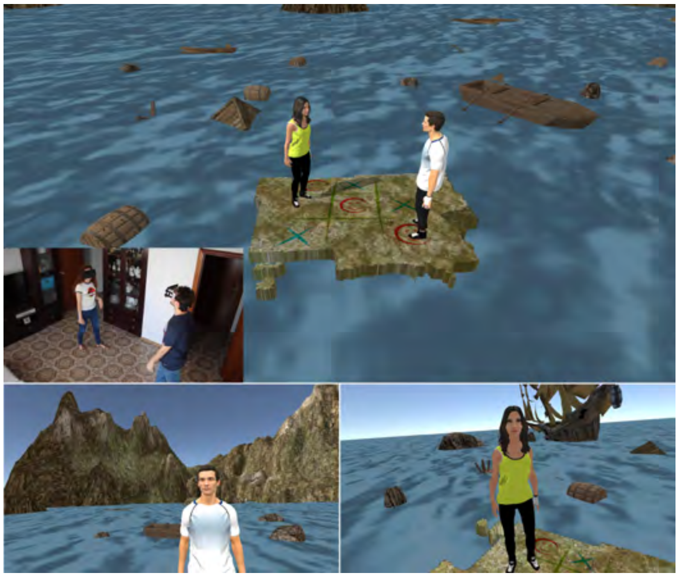

Fig. 11. Multiuser scenario (Fig. 4i) where two users share a single RE. Top image shows two users/avatars playing tic-tac-toe and the position of both the users in the RE are shown in the inset. Bottom two images show the views from a first person perspective of each user.

Scenario 1) is the most interesting as it allows geographically distributed people in completely different types of physical environments to interact together in a single and common VE which is generated automatically. A specific case is shown in Figure 10 where each user is placed in an independent part of the virtual world so that each person has their entire WA with no overlap with the other person. The users can still see each other and interact to some degree. Another scenario, presented in Figure 9 shows two WAs overlapping to afford closer interaction.

In our implementation of Scenario 3), the non-VR user

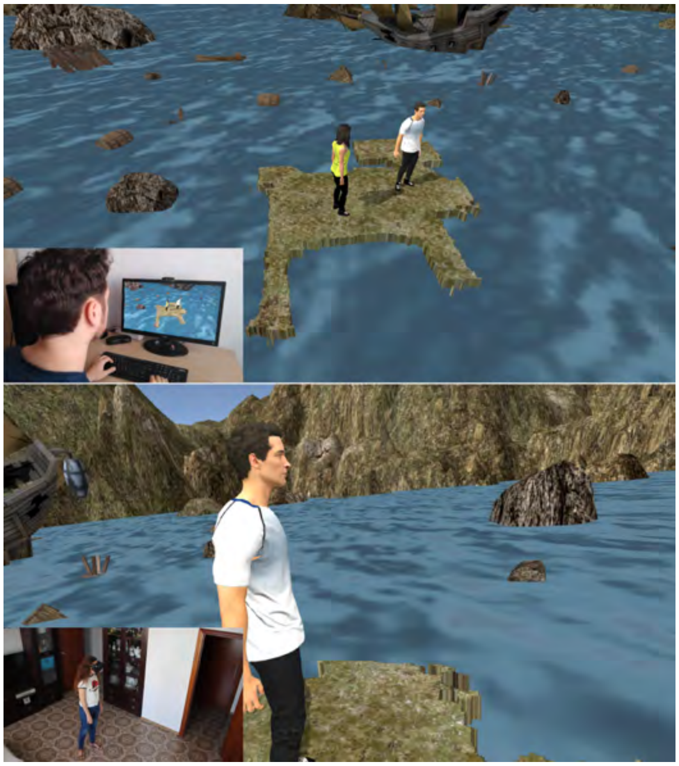

Fig. 12. Example of asymmetrical multiuser scenario. Male avatar user is connected through a computer, as shown in top image inset, while the female avatar user is connected using a Tango tablet worn as an HMD, as shown in the bottom image inset. The RE corresponds to 4a. In this case there is no need of a WA combination as only one RE is used.

is connected using a PC similar to playing a standard video game. We allow both, first and third person perspectives for the PC user. For the VR user this is transparent as they cannot distinguish between participants connected using a VR device or a PC. The main difference in this case is that there is no need to generate a combined WA since only the VR participant will have a WA and not the seated PC user. This setup allows for more people to participate in a shared VR experience without needing special VR equipment.

In addition to sharing a VE, users can also interact with one another. In the VE, each user is assigned a virtual avatar as shown in Figure 9 to indicate their position and orientation to the others. This avatar moves in the VE as the users walk in their REs (or is moved with the keyboard in the asymmetrical case). The avatar automatically synchronizes with the user real world movements, rotating according to Tango's orientation and displaying walking animation when users change their position in the RE.

Participants can interact with one another through their relative positions and orientations in the VE, even when they are in different physical spaces. Pose data allows the detection of specific actions or properties such as the distance between two users, when they look to each other or when a user is standing still or walking. A combined WA along with user pose data opens up the possibility of creating collaborative experiences such as virtual meeting rooms for teleworkers which automatically adapt to the different REs of each worker, or custom distance learning classes, collaborative painting, storytelling etc. Another big opportunity is the creation of multiplayer games where the position and orientation of each user is employed for interaction with the environment and with other players. Examples can vary from simple tower defense games to something more complex such as team-based rail shooters or RPGs in the combined WA. We demonstrate using pose data for playful interactions through tic-tac-toe played on the ground by two users (Fig. 11).

5.2 Physical Object Interaction

In Figure 13b, we use a virtual chair that is similar in style to the rest of the elements in the generated virtual world. Because it is not a replica of the physical chair, its virtual and physical properties differ from an aesthetic and tactile perspective. The altered physical properties do not affect the way the objects functionality is perceived [21]. For example, in the VE, the material of the chair may appear to be made of wood or stone. This affects the way the object is perceived in terms of temperature, weight, texture, and hardness but not functionally; the user still expects to be able to use the chair for sitting. The visual discrepancy can be resolved by using a high-fidelity chair model that more closely matches the real chair. Prior work shows as long as the discrepancy between the visual information and the haptic information does not get too large, the visual information will dominate [32] and the discrepancy will not adversely impact the user's experience in virtual reality. In the chair example in Figure 13b, the user gets a more engaging tactile experience of sitting in a virtual chair while simultaneously sitting is the matching physical proxy. The example scene of a chair on a floating raft may seem contrived but the goal of that specific scene was to reiterate that obstacles like tables and walls in the room, that are outside the WA, are represented by a boundary element, e.g., water and not a virtual counterpart. Only the selected physical objects in the room have a surrogate virtual object with which the user can interact.

The presence of virtual representations of real world objects does not limit the flexibility of the generated VE as long as the number of detected and interactive objects is kept low. This low limit allows the design of interactions with some specific objects while still maintaining the idea of the virtual world being totally different from the real world.

It must be noted that since the user can get close to a detected interactive object, like the chair in Fig. 13b, the object needs to be included in the WA segmentation. We include the chair in the area by projecting the bounding box of the detected chair onto the estimated floor plane image. To avoid user collisions, interactive objects that are too close to obstacles need to be ignored (e.g., a pushed in chair). To do so we can consider different factors such as distance between potential interactive object and obstacle, required space to interact, object and obstacle heights, etc.

5.3 Virtual Object Interaction

Some virtual elements in the VE respond to the user's presence, making the world feel alive. For example, in the sample world shown in Figure 13a, virtual animals flee when the user approaches or startles them. Though we demonstrate only a few, triggers based on changes in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2762691, IEEE Transactions on Visualization and Computer Graphics

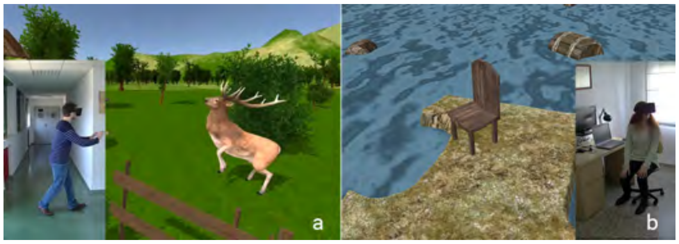

Fig. 13. User interactions in the virtual world. (a) Proximity based interaction with virtual elements (virtual world as seen from a first person perspective). (b) Interaction with a real chair through passive haptics (virtual world as seen from a third person perspective).

user's position and orientation can be easily included for a richer virtual experience. Possible triggers include, detecting if the user is looking at the floor, if they are bending or running, or if they have entered a particular area.

6 EVALUATION AND DISCUSSION

6.1 Implementation Details

The point cloud analysis, object detection and virtual world generation process was done on a Windows 10 computer with an Intel Core i7-6700K 4.0GHz processor and 16GB RAM. Object tracking was done on the mobile device. The virtual world was generated using the Unity game engine and textures, models and other 3D elements were downloaded from the Unity Assets Store.

Total time required to process a point cloud and generate a virtual environment was about 2-4 minutes for each physical space we tested, including the time taken to do the initial 3D mapping. A time consuming pre-processing task was object recognition, which varied from 20 minutes to 1 hour depending on the complexity of the scene and the object being detected. This performance can be improved drastically by reducing the large set of instances we test against, 880 in case of the chair, though that may reduce accuracy, which is necessary for sitting down. When we tested with 20-40 instances, the object detection time came down to 2 minutes.

We determined that since object recognition would be done once at the beginning, it would not impact the VR experience adversely by not running in real-time. However, keeping the physical object and its virtual counterpart in sync during VR mode was needed to avoid collisions and thus real-time object tracking was implemented. We achieved a frame rate of 20-25 fps during immersion with one object being tracked.

The GA employed a population of 1000 individuals, a mutation probability of 0.05 and a maximum number of 1000 iterations. These parameters, along with the weights w_i of the fitness function were determined experimentally in order to fulfill the proposed rules.

We use the Tango tablet both as the 3D mapping device and as an HMD. It is self contained, mobile, provides low latency tracking in real-time, has higher computational resources than other similar tablet devices, and has integrated sensors that are necessary for accomplishing our goals. The Tango SDK provides functionality to perform 3D reconstruction and user tracking. The Constructor[8] tool for Tango was used for 3D reconstruction in our examples. All other steps, i.e., WA detection, VE generation, object recognition and tracking, were specifically implemented for this work. Any other HMD which allows inside-out user tracking could be used for the immersive experience. Since all we need is a standard point cloud of the physical environment, data from non-HMD devices like the Microsoft Kinect or the Intel RealSense camera would be acceptable for generating the VE. However, for object tracking, it would be necessary for the HMD to have a depth sensor in order to apply ICP in real-time, for our implementation.

In the multiuser system, each RE is scanned in the same way as in the single user system, using a Tango tablet device. The combined WA is estimated during the analysis of the point clouds and it takes about 2-4 seconds to calculate the combined WA for two REs. During immersion, each user shares their pose in the VE with everyone else over the network. For the asymmetrical case, a standard PC is employed.

6.2 Qualitative analysis

We evaluated our system in different real environments and with several different users to test its functionality and capabilities. One evaluation task of paramount importance was safety of the system as users walk around while their senses of sight and sound are occluded. When the tiered warning system was disabled, some users ignored the boundary elements, as we expected, and had to be steered away from the physical obstacles. When the tiered warning system was enabled, those who ignored the boundary elements successfully managed to steer themselves away from the boundary, without external help. There were some minor collisions due to users reaching out with their arms that could not be prevented as we do not implement hand tracking.

For the tiered warning system, we consider the case of a user walking forward so it is only activated for obstacles that are at front of the user, not at the back or sides. During the pilot, users tended to step back when the warning was activated, especially the audio warning. This is why we chose to activate the warning system for obstacles in front of the user otherwise they may accidentally run into something if the warnings come from other directions. One user mistook the visual warning system for a game feature. Another user found it distracting.

Qualitative user feedback was positive and almost invariant. Test subjects were very excited, which is possibly caused by the novelty effect combined with a high level of presence in the fully immersive VR system. All subjects were tentative about sitting in the virtual/physical chair and expressed surprise on being able to do so. In more than 20 tests (with 10 participants), no subjects reported any nausea during or after the experience. This surprisingly good result may be explained by the lack of conflict between visual and vestibular cues.

In Figure 14 we present the results using our approach for three different physical environments. Each scene is a

[8] Tango Constructor. https://github.com/lvonasek/tango

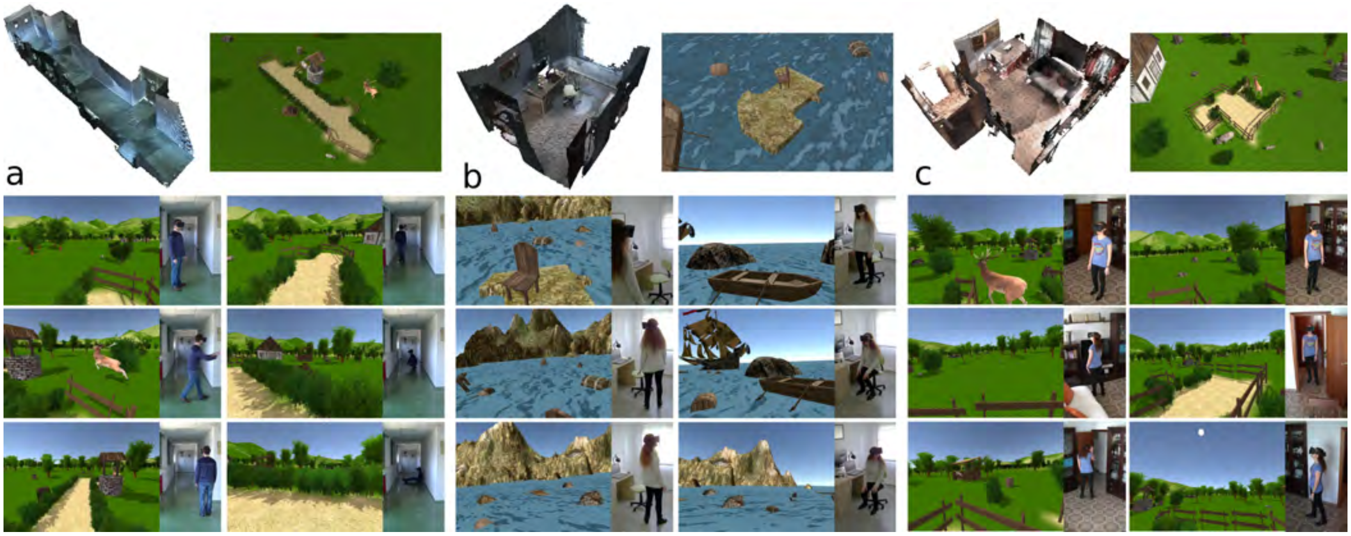

Fig. 14. Examples showing user immersion in three different real environments. (a) Hallway. (b) Office. (c) Living room.

visually-immersive real-walking VE that the user experiences through an HMD. The user can move, walk, bend down in the environment while their pose (position, orientation) is continuously tracked by the Tango device. Figure 14b shows interaction with a real chair.

The ability to generate larger than room-scale VEs and walk freely in them provides a significant improvement compared with existing commodity virtual reality systems, such as HTC Vive. By using boundary elements that perceptually imply a border, e.g., fences, the user clearly understands the demarcation between the space where they can safely move and the rest of the environment that is physically out of bounds but still accessible visually.

6.3 Multiuser qualitative analysis

We designed an initial prototype of our multiuser system and ran a pilot study with four users in order to get feedback on social interaction as well as to find approaches for future improvements. We asked the participants to perform a task that required them to be in physical proximity in the VE. We did not include any detected chairs for the pilot as we wanted to focus on interaction between the users. Each user tried all three scenarios namely, 1) both users in the same physical space, 2) both users in different physical spaces, and 3) one user in VR and the other user connected through a PC. In general, reactions were positive. People appreciated being able to walk towards another user to get closer to them. They felt like they were standing next to each other, especially in scenario 2). Two users remarked that being able to see the full body of the other user helped them be aware of each other's personal space and also felt more realistic. All users expressed a desire to control their avatar's body movements, especially hands.

A primary concern that emerged from the feedback was the restricted movement space. This was due to the small sizes of the scanned physical spaces leading to an even smaller combined WA. This problem was specially noticeable for PC users as they could explore the full VE faster than a VR user by using the keyboard to move. This is indeed a limitation but can be mitigated by scanning the entire home/apartment/office instead of one room to expand the WA for each user.

Overall, sharing the virtual world with another user was appreciated and people enjoyed the experience. For communication, in scenario 1) users were able to talk naturally since they were in the same physical space. For scenario 2) and 3) we setup a Discord session to allow for voice chat. Participants played tic-tac-toe (see Fig. 11) by moving their body to the different spaces in the 3x3 grid. The tic-tac-toe board was automatically placed on the floor of the WA and resized to fit as needed. All participants enjoyed this interaction and commented that it reminded them of their childhood games.

6.4 Limitations

Our system has some limitations that are worth mentioning. The Tango device provides position and orientation data that is used in both the 3D reconstruction and motion tracking phases. In some cases, incorrect estimation of the user's pose can generate a displacement between their real and virtual world positions. The drift usually appears when the motion tracking system does not have reliable data to perform correct tracking, for example, because of fast movements, dimly lit spaces, relatively empty spaces with no features like blank walls and floors, or proximity of the device to obstacles such that camera images are not available. In the multiuser scenario where the RE is shared, we observed some tracking problems with the Tango device due to occlusions caused by the users bodies as well as sensor interference between the two Tango devices. Communication between the two users is important, if they are in the same RE, to prevent colliding with each other.

A user's sense of presence in the VE may be disrupted if they accidentally or purposefully touch the boundary elements. Since the boundary elements separate the WA from the obstacles in the physical space, touching them would mean touching a wall or a piece of furniture. However, since the boundary elements are not virtual counterparts of these real world objects, the disconnect between touching a virtual shrub and a physical table may be jarring. We chose not to include virtual counterparts of all physical objects because we were interested in generating large open virtual spaces for the available indoor spaces instead of limiting ourselves to generating virtual spaces that have a 1:1 mapping with the available physical space, especially since that has been explored by [16].

Currently, we do not track other people or pets in the RE who could collide with or startle the VR user. We expect individuals not wearing an HMD to easily avoid colliding with the person wearing an HMD but pets may need to be tracked or temporarily removed from the space. Since one of the first steps in WA detection is estimating the planar floor in the environment, the proposed system is limited to environments that do not include, ramps or stairs. Contrary to Vive and other VR systems, Oasis allows creating VEs from large REs such as full apartments. This works well for the single user experience. However, in the multiuser case, the combined WA is usually smaller compared to the individual WAs as it is based on the overlapped area between the two. The final size of the combined WA depends on the specific shape of each individual's WA.

7 Conclusion and future work

In this paper, we described a novel VR generation system that uses the real world as a template and allows natural walking in the generated virtual world and incorporates haptic feedback. The system, thus, creates a highly immersive environment by combining egocentric scene viewing with proprioception, vestibular, and tactile feedback.

Our system is the first to allow casual users to quickly and easily create immersive and interactive VEs that can be experienced in an HMD. In order to achieve our design goals, we devised a procedural virtual world generation framework using 3D reconstruction and object recognition data. We demonstrated the procedural generation through the automatic creation of a variety of virtual scenes for the same physical environment. Besides overcoming the previously presented limitations, there are a few aspects of the system that we can improve upon in future work.

An immediate improvement would be adding a teleportation system for moving the combined WA to new locations in the virtual environment (as in [33]). While new locations would be restricted to only those places in the VE where the WA fits, the added teleportation would allow users to travel and explore much larger virtual spaces using relatively small physical spaces available for walking. To create the illusion of a larger combined WA, we can employ narrative and interaction design techniques such as virtual metaphors to delimit areas that can only be reached by specific users, e.g., doors that only open to certain users. This would allow users to seamlessly share parts of the VE corresponding to combined WAs while still being able to walk freely in their individual WAs. Similarly, metaphors can also be employed to share interactive physical objects, such as chairs, in multiuser scenarios even if the object is present in only one of the REs. For example, only one participant would be allowed to interact with the tracked object, e.g., a red chair can only be used by the red avatar.

User interaction can be improved by attaching a hand detection sensor device such as Leap Motion to each Tango Device. We believe including hand tracking will enhance the experience and allow for a larger set of interactions between users and between the users and the VE that go beyond proximity and orientation based interactions. This idea can be extended by adding a tracked full-body avatar that moves in sync with the user's real body movements. This may require using either a motion capture suit or some external tracking device like the Kinect or Vive though that would limit the size of the available walkable area.

To reduce user collisions in multiuser scenarios, a partial solution could be to implement a personal bubble (similar to Facebook Spaces) around each avatar that prevents others from coming close virtually. Prior research shows that real world proxemics behaviors work the same way in VR [34], [35]. Participants move out of the way when approached by virtual avatars and keep greater distances when there is mutual gaze [36]. As long as users match their physical behavior with their virtual behavior, the personal bubbles could successfully minimize awkwardness due to collisions.

Other important potential improvements to optimize system performance and subjective experience are: (i) Expanding object detection and tracking to a wider range of objects beyond furniture and potentially to other people and pets. (ii) Employing more sophisticated procedural map generation techniques for creating the virtual world. (iii) Providing predefined theme-based sets of 3D models (e.g., space, forest, fantasy) for giving the user an option to choose the theme for the generated VE. (iv) Real-time VR generation, where the virtual world unfolds and layers itself over the real world as the user walks around wearing an HMD.

References

[1] I. E. Sutherland, "A head-mounted three dimensional display," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 1968, pp. 757–764. 1

[2] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The cave: audio visual experience automatic virtual environment," *Communications of the ACM*, vol. 35, no. 6, pp. 64–73, 1992. 1

[3] G. Burdea and P. Coiffet, "Virtual reality technology," *Presence: Teleoperators and virtual environments*, vol. 12, no. 6, pp. 663–664, 2003. 1

[4] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected walking," in *Proceedings of EUROGRAPHICS*, vol. 9. Citeseer, 2001, pp. 105–106. 1, 2

[5] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr, "Walking > walking-in-place > flying, in virtual environments," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 359–364. 1, 2

[6] T. Field and P. Vamplew, "Generalised algorithms for redirected walking in virtual environments," 2004. 1

[7] B. E. Insko, "Passive haptics significantly enhances virtual environments," Ph.D. dissertation, University of North Carolina at Chapel Hill, 2001. 1, 2, 3

[8] H. G. Hoffman, "Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments," in *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*. IEEE, 1998, pp. 59–63. 1, 2

[9] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE ISMAR*. IEEE, October 2011. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=155378 2, 8

[10] B. Peasley and S. Birchfield, "Replacing projective data association with lucas-kanade for kinectfusion," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 638–645. 2

[11] U. Qayyum, J. Kim *et al.*, "Inertial-kinect fusion for outdoor 3d navigation," in *Australasian Conference on Robotics and Automation (ACRA)*, 2013. 2

[12] A. Nassani, H. Bai, G. Lee, and M. Billinghurst, "Tag it!: Ar annotation using wearable sensors," in *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, ser. SA '15. New York, NY, USA: ACM, 2015, pp. 12:1–12:4. 2

[13] K. Lai, L. Bo, X. Ren, and D. Fox, "Rgb-d object recognition: Features, algorithms, and a large scale benchmark," in *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 167–192. 2

[14] L. A. Alexandre, "3d descriptors for object and category recognition: a comparative evaluation," in *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*, vol. 1, no. 3. Citeseer, 2012, p. 7. 2

[15] S. Song and J. Xiao, "Sliding shapes for 3d object detection in depth images," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, 2014, pp. 634–651. 2, 8

[16] M. Sra and C. Schmandt, "Metaspace: Full-body tracking for immersive multiperson virtual reality," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 47–48. 2, 3, 13

[17] T. Nescher, M. Zank, and A. Kunz, "Simultaneous mapping and redirected walking for ad hoc free walking in virtual environments," in *IEEE Virtual Reality Conference 2016*. IEEE, 2016, pp. 239–240. 2

[18] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte, "Leveraging change blindness for redirection in virtual environments," in *2011 IEEE Virtual Reality Conference*. IEEE, 2011, pp. 159–166. 2

[19] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell, "Passive real-world interface props for neurosurgical visualization," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1994, pp. 452–458. 2

[20] L.-P. Cheng, T. Roumen, H. Rantzsch, S. Köhler, P. Schmidt, R. Kovacs, J. Jasper, J. Kemper, and P. Baudisch, "Turkdeck: Physical virtual reality based on people," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 417–426. 2

[21] A. L. Simeone, E. Velloso, and H. Gellersen, "Substitutional reality: Using the physical environment to design virtual reality experiences," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 3307–3316. 2, 10

[22] A. Hettiarachchi and D. Wigdor, "Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 1957–1967. 2

[23] L. Shapira, R. Gal, E. Ofek, and P. Kohli, "FLARE: fast layout for augmented reality applications." IEEE Institute of Electrical and Electronics Engineers, September 2014. [Online]. Available: https://www.microsoft.com/en-us/research/publication/flare-fast-layout-for-augmented-reality-applications/ 2

[24] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson, "Snaptoreality: Aligning augmented reality to the real world," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 1233–1244. 2

[25] R. C. Waters, D. B. Anderson, J. W. Barrus, D. C. Brogan, M. A. Casey, S. G. McKeown, T. Nitta, I. B. Sterns, and W. S. Yerazunis, "Diamond park and spline: Social virtual reality with 3d animation, spoken interaction, and runtime extendability," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 461–481, 1997. 3

[26] M. Sra, D. Jain, A. P. Caetano, A. Calvo, E. Hilton, and C. Schmandt, "Resolving spatial variation and allowing spectator participation in multiplayer vr," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 221–222. 3

[27] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994. 7

[28] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 2–9. 7

[29] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606. 8

[30] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014. 8

[31] J. J. Gibson, *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. 9

[32] D. H. Warren and W. T. Cleaves, "Visual-proprioceptive interaction under large amounts of conflict." *Journal of experimental psychology*, vol. 90, no. 2, p. 206, 1971. 10

[33] D. A. Bowman, D. Koller, and L. F. Hodges, "Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques," in *Virtual Reality Annual International Symposium, 1997., IEEE 1997*. IEEE, 1997, pp. 45–52. 13

[34] A. Guye-Vuillème, T. K. Capin, S. Pandzic, N. M. Thalmann, and D. Thalmann, "Nonverbal communication interface for collaborative virtual environments," *Virtual Reality*, vol. 4, no. 1, pp. 49–59, Mar 1999. 13

[35] L. M. Wilcox, R. S. Allison, S. Elfassy, and C. Grelik, "Personal space in virtual reality," *ACM Trans. Appl. Percept.*, vol. 3, no. 4, pp. 412–428, Oct. 2006. 13

[36] J. N. Bailenson, J. Blascovich, A. C. Beall, and J. M. Loomis, "Interpersonal distance in immersive virtual environments," *Personality and Social Psychology Bulletin*, vol. 29, no. 7, pp. 819–833, 2003. 13

Misha Sra is a Ph.D. candidate in the Fluid Interfaces Group at the MIT Media Lab where she focuses on redefining the boundary between the real and the virtual. She works at the intersection of virtual reality, machine learning and computer vision exploring the design of novel ways to easily and automatically create personalized 3D virtual environments. Her research interests are natural movement techniques, embodiment, and collaboration in VR and AR.

Sergio Garrido Jurado received his Ph.D. degree in Computer Vision from the University of Córdoba (Spain) in 2016. During the last years he has collaborated as a researcher with the AVA Research Group at the University of Córdoba and the Fluid Interfaces Group at the MIT Media Lab. He has also been involved in several commercial augmented and virtual reality products. His research interests are in the areas of SLAM, object tracking and 3D reconstruction.

Pattie Maes Pattie Maes is a professor in MIT's Program in Media Arts and Sciences and head of the Program in Media Arts and Sciences. She founded and directs the Media Lab's Fluid Interfaces research group. Her areas of expertise are human-computer interaction and artificial intelligence. She is particularly interested in the topic of human augmentation, or how systems can actively assist people with memory, learning, decision making, communication and physical skills.