# TOWARDS GESTURE-BASED MULTI-USER INTERACTIONS
# IN COLLABORATIVE VIRTUAL ENVIRONMENTS

Nicolò Pretto[a] and Fabio Poiesi[b,*]

[a] Department of Psychology and Cognitive Science, Universitá degli Studi di Trento - info@npretto.com
[b] Technologies of Vision - Fondazione Bruno Kessler - Povo (TN), Italy - poiesi@fbk.eu

**Commission II**

**ABSTRACT:**

We present a virtual reality (VR) setup that enables multiple users to participate in collaborative virtual environments and interact via gestures. A collaborative VR session is established through a network of users that is composed of a server and a set of clients. The server manages the communication amongst clients and is created by one of the users. Each user's VR setup consists of a Head Mounted Display (HMD) for immersive visualisation, a hand tracking system to interact with virtual objects and a single-hand joypad to move in the virtual environment. We use Google Cardboard as a HMD for the VR experience and a Leap Motion for hand tracking, thus making our solution low cost. We evaluate our VR setup though a forensics use case, where real-world objects pertaining to a simulated crime scene are included in a VR environment, acquired using a smartphone-based 3D reconstruction pipeline. Users can interact using virtual gesture-based tools such as pointers and rulers.

## 1. INTRODUCTION

Virtual Reality (VR) has innovated the way we analyse, interact with and enjoy 3D data through the creation of immersive visual experiences. Users can experience immersive VR through wearable head mounted displays (HMDs) that perform an egocentric encoding of the scene, whereas allocentric-encoded scenes displayed through computer screens lead to non-immersive VR experiences (Kozhevnikov and Gurlitt, 2013). VR also offers more creative visual interfaces that lend themselves to analytical reasoning by the richer exploitation of human perception, intuition and pattern recognition, thus enabling users to interact with objects in forms that would not normally be possible in the real world (Olshannikova et al., 2015) (Butcher et al., 2016). The design and development of computer games, arts, fashion, digital storytelling and advertising are just a few of the application domains where immersive VR is already exploited. Thanks to improved rendering techniques, more accurate inertial sensors and natural interactive tools, VR has also begun to be employed in therapy clinics (Spicer et al., 2017), architecture (Vorlander et al., 2015), urban planning (Carrozza et al., 2014), museum tours (Carrozzino and Bergamasco, 2010), education (Greenwald et al., 2017a) and forensic investigations (Ebert et al., 2014). In forensics for example, crime scenes are nowadays routinely replicated in 3D, and VR can be a cost-effective solution to observe a scene from various viewpoints, such as a victim's view and a witness' view (Burton et al., 2005). Computer animations can be used for evidence presentation to juries and judges in courtrooms (Ma et al., 2010) (Ebert et al., 2014). Experts of different forensics departments with different domains of expertise can also collaborate within the same VR environment while being geographically distant from each other. These collaborative virtual environments (CVEs) have shown to be a very effective means of stimulating collective thinking, supporting the generation of ideas and contributing to joint data understanding (Churchill and Snowdon, 1998) (Greenwald et al., 2017b). Although CVEs have been studied for over two

decades, collaborative visual analytics have not yet been largely tested in real use scenarios.

Immersive VR has just recently become popular thanks to affordable VR hardware solutions. A typical dedicated VR setup is composed of a HMD that is connected to a computer via HDMI to receive visual graphics information and via USB to send sensory data (i.e. gyroscope, accelerometer, magnetometer), respectively. Popular HMDs are HTC Vive, Oculus Rift, PlayStation VR, Gear VR and Google Cardboard (or simply Cardboard). The first three solutions do not have embedded processing units and need a wired link to a computer to process the data. Although there are mechanisms to make the link wireless using mmWave multi-Gbps data rates, the HMDs still need to be in the antenna's line-of-sight due to the large bandwidth requirement, and thus occlusions can cause communication interruptions (Abari et al., 2016). Gear VR and Cardboard use smartphones as fully-embedded HMDs, thus they do not require a wired connection to a computer. Due to the limited processing capability of smartphones, both Gear VR and Cardboard offer lower quality renderings compared to dedicated options. Despite this, the ubiquitousness nature of smartphones, along with their increased computational power and ease to build applications, is contributing to making VR accessible to almost anyone (Castelvecchi, 2016). Moreover, advances in computer vision algorithms together with high-quality embedded cameras are enabling smartphones not only to be used as VR HMDs, but also to become a bridge between VR and the real world. Users are able to reconstruct real-world objects in 3D, manipulate their reconstructed models in VR space, and then share them in a shared environment across different users, entirely on their smartphones. This fast-growing technological progress is leading to new applications and ways we create, access and share digital content, stimulating interconnectivity activities in multi-user VR spaces.

In this paper, we present a system to enable multi-user interactions via hand tracking devices to promote collaborative VR analytics (Fig. 1). Communications across HMDs is achieved using
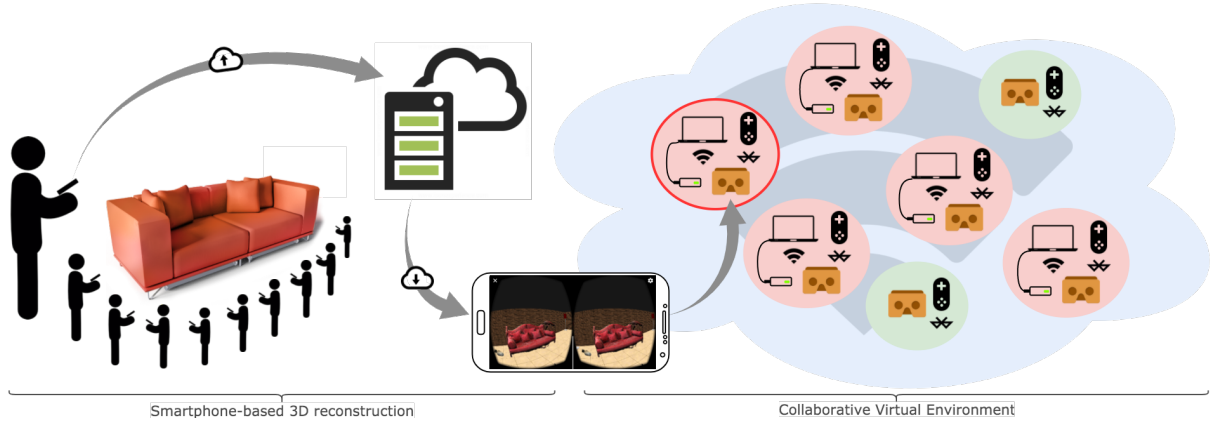
---

*Corresponding author.

Figure 1. Diagram of a collaborative VR setup comprised of a smartphone-based 3D reconstruction step (left-hand side) and an immersive collaborative experience (right-hand side). Automatically selected images from a smartphone's camera feed are sent to a reconstruction server that produces the 3D model of a scanned object. This 3D model can be imported in an immersive VR environment where multiple users can collaborate. The collaborative virtual environment is based on a client-server architecture. The server is represented as a red contour. The user that starts a VR session is the server. Clients connect to the server either as active or passive users, red and green colours on the right-hand side, respectively. A Leap Motion (orange VR icon inside each circle) is connected to each computer and hand-tracking data is transmitted to an associated Cardboard device. Users can interact with virtual objects in the same VR space and see each other's actions.

WiFi, enabling users to collaborate inside the same virtual environment without necessary being in the same physical space. Our presented system is based on Cardboard and Leap Motion. Leap Motion is an optical device that, by connecting it to an external processing unit, can perform hand and finger tracking. Despite modern smartphones' powerful processor capabilities, the Leap Motion SDK does not yet support a direct connection between smartphones and Leap Motion devices. We have bypassed this limitation by connecting the Leap Motion to a computer and streaming hand-tracking data to smartphones via WiFi. To the best of our knowledge, our approach is the first work that enables multiple users to collaborate within a shared virtual environment through gesture-based interactions using Cardboard. We evaluate the proposed system in a forensics scenario by simulating a crime scene that contains virtual objects mixed with 3D models that have been reconstructed using a smartphone app. In our VR scenario, participants can interact with each other and virtual objects by using virtual rulers and pointers. We analyse bandwidth usage as a function of connected users using off-the-shelf networking hardware and smartphones.

## 2. OVERVIEW

The presented system is composed of a server that hosts a VR session and a set of clients that join this session. We use a server authoritative system, which allows one of the users to be concurrently a client and the server, in this way no dedicated server process is required. Each user can set up a VR session and enable others to join it as a participant. Each user is represented as a character with a distinctive colour and can actively participate in a VR session by interacting with objects via their Leap Motion device or by moving inside the virtual environment using a joypad. A user can also passively participate in a VR session and only watch others' interactions. Hand tracking data and character movements are sent to the server that in turn broadcasts them to the other clients. In this way all users can see each other's actions. The visual responsiveness across participants depends on network communication performance. Fig. 1 shows the design of our proposed communication system.

## 3. SINGLE-USER VR INTERACTION

### 3.1 Hand tracking

The setup of an active user consists of a Cardboard HMD for immersive visualisation, a Leap Motion for hand tracking and a joypad to move one's character inside a virtual environment. The Leap Motion device embeds two monochromatic IR cameras and three infra-red LEDs for hand and finger tracking. Video frames captured by the Leap Motion are transmitted to a computer via USB. A computer processes received video frames to estimate the positions and orientations of both hands and fingers at a frame rate between 20Hz and 200Hz, based on the computers' specifications. In this paper, we have named the tracking data generated by the Leap Motion of a certain frame instants as *leap-frame* (Leap-frame, 2017). Leap-frames are sent from the computer to the Cardboard in a serialised format. They are then de-serialised by the Cardboard device for visualisation. The transmission of leap-frames is performed via WiFi using the UDP protocol to ensure timely delivery. As opposed to TCP, UDP does not perform error control on the transmitted data, therefore we make use of MD5 hashes to validate the quality of transmission. On the computer, we compute an MD5 hash (128 bits) for each leap-frame, which is subsequently appended to its corresponding leap-frame and then it is transmitted to the Cardboard. When the Cardboard receives a leap-frame, it computes the MD5 hash of the leap-frame and checks to see if it is the same as that of the appended MD5 hash. Only leap-frames with the same MD5 hash on transmission and reception are visualised on the display of the smartphones. To ensure the timely visualisation of leap-frames on Cardboard, we only visualise the latest received leap-frame, discarding those delayed or corrupted. The high frame rate guarantees a smooth visualisation of the tracked hands and fingers, even though some leap-frames may be discarded.

Fig. 2 shows a single-user VR setup during a Research Open Day of the ICT centre in Fondazione Bruno Kessler. The laptop and Cardboard are connected via WiFi to a router using the *eduroam* network (EDUROAM network, 2017). Software on the computer enables the user to select which device leap-frames are transmitted,

Figure 2. A single user experiencing VR with Leap Motion connected to Cardboard via WiFi. The picture was taken during the Research Open Day of the ICT centre in Fondazione Bruno Kessler. Although the communication between the two devices relied on the main router in the room with several people connected to the same network, the users experienced a smooth and delay-free hand-based interaction with virtual objects.

i.e. the Cardboard in our case. Although there were dozens of people connected to the WiFi network during the event, users could still experience a smooth and delay-free gesture-based interaction with virtual objects.

## 3.2 Gesture-based interactions

We associated gestures to actions to enable a user to interact with the virtual environment. In this work, we encoded three actions: pinch, point and measure.

The *pinch* actions let a user pick up objects and then move them around. A pinch is detected when the thumb and index finger get close to each other. If this action occurs near a "pinchable" object, the object will be picked up and it will follow the position of the pinching point. When the fingers are separated, the object will be released. Fig. 3a shows an example where a knife is picked up.

The *point* action is activated when a user closes all of their fingers except for the index finger. This gesture generates a ray that starts from the tip of the index finger and then terminates on the first surface hit by the ray. This feature is useful when a user wants to highlight an object to another users in the VR environment. Fig. 3b shows an example where the index finger points the floor and a white ray appears.

The *measure* action uses the point action to place markers in positions where rays hit surfaces. In particular, a line will be generated between two placed markers, and the value of the distance will be visualised on the top of this line. Fig. 3c shows an example of the ruler used to measure the distante between two markers.

## 4. GESTURE-BASED COLLABORATIVE VR

Collaborative VR is achieved through multiple Cardboard devices that communicate via WiFi. Each client sends gesture and character movement data to the server. The server then broadcasts this data to the other users.
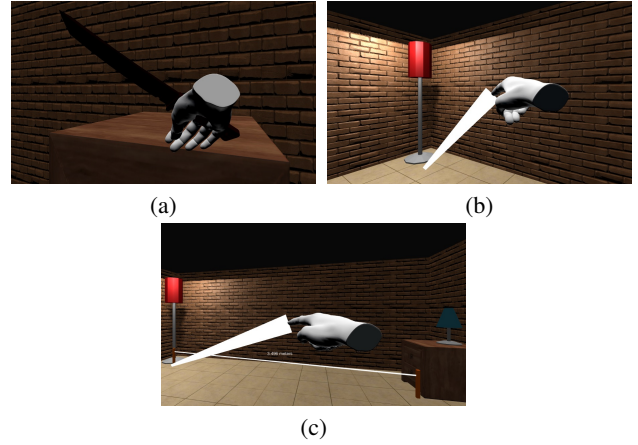


(a)       (b)



(c)

Figure 3. Gestures associated to the actions of (a) pinch, (b) point and (c) measure. Pinch allows users to pick up objects. Point allows users to highlight objects. Measure allows users to measure distances between virtual objects.

### 4.1 Server-clients communication

The design of our server-client architecture is based on Unity 3D (Unity 3D, 2017) and has been developed with their *High Level API* (HLAPI) (High Level API, 2017). HLAPI provides an interface that is built on top of a real-time transport layer that manages the task of multiple user interaction in the same VR environment. When a user runs a VR session, our system offers two options: *Host* or *Join*. The *Host* option creates a new VR session that is visible in the local network and it appoints that user's Cardboard as the server. Visibility in the local network is achieved by broadcasting a message (e.g. "broadcast search") at regular intervals. The *Join* option checks to see if there is an ongoing VR session by listening for this message in the local network. When this message is read by the client, the Cardboard automatically joins the session. Otherwise, if no message is read within a certain period of time, the search will go in *time out* mode, and the Cardboard will notify the user that there are no active sessions in the local network. When a user joins a VR session, a new character with a unique identity (ID) and colour will spawn in the shared VR environment.

Inside a VR session, all user actions are sent through the network by Remote Procedure Calls (RPCs), which are part of HLAPI (High Level API, 2017). Calls are used to update the state of objects and characters, e.g. the position of the characters and of the hands/fingers, so that all participants can be visualised in the same VR environment. RPCs come in two types: Command and ClientRpc calls. *Command* calls are used when a client wants to perform an action that needs to be visible by other participants. This type of call is generated on the client and runs on the server. *ClientRpc* calls are instead called on the server and run on the clients. In order to have a smooth and delay-free visualisation for all participants, these calls should be transmitted and processed at least 30 frames per second. For example, when a user moves their character, information about the character's state (e.g. position) in 3D space is sent to the server via a Command call. The server updates the state of this character and broadcasts the update with a ClientRpc call. When a client receives this update, the character's state will be updated and its movement will be displayed on their HMD. If these network calls undergo delays, clients will observe jerky character movements. The same principle applies to hand state updates, where hand and finger positions and orientations for all participants need to be refreshed. However, as opposed
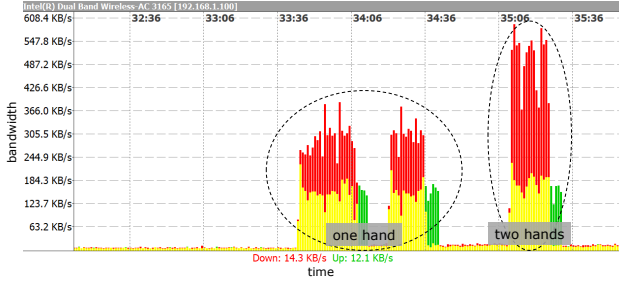
Figure 4. Bandwidth that is measured on a Cardboard client to download leap-frames (red bars) from a computer and to upload leap-frames (green bars) to the Cardboard server in the case of one hand and two hands tracked. Yellow bars represent the overlap between the download and upload.

to character positions, hand visualisations requires updates at a higher frame rate due to the large number of joints and degrees of freedom the Leap Motion tracks, and it even increases if multiple participants concurrently move their hands. We mitigate this problem by implementing a scheduling strategy on the server to broadcast updates that selects one ID at a time at regular intervals. The scheduling strategy is based on a sequential circular counter that increments every time an update is sent. Command calls of users that do match the selected ID are not broadcasted. In this way, we can ensure that the server broadcasts network calls at regular intervals based on the total number of participants, thus avoiding transmissions of large packets, call queues and delays. When a client receives a ClientRpc call from the server, it also contains the leap-frame of the selected ID from that instant. The client then de-serialises the leap-frame and updates the state of the hands belonging to that character ID.

### 4.2   Bandwidth analysis

We tested a collaborative scenario with three users connected to the same local network via WiFi. One active user (the server) hosted a collaborative VR session. Two users (the clients) then joined the collaborative VR session as active and passive users, respectively. The active users were connected to their personal Leap Motions using separate computers.

Fig. 4 shows the bandwidth utilised by a Cardboard client. From the graph we can observe that the bandwidth used by the Cardboard to download leap-frames from the computer is proportional to the number of hands, that is ∼300KB/s for one hand and ∼600KB/s for two hands. Conversely, the bandwidth used to upload the leap-frames of one hand and two hands to the server is the same and about 150KB/s. The difference between download and upload bandwidths is due to our implementation on the Cardboard. The leap-frames received from the computer with the Leap Motion are managed with an ad-hoc code that guarantees a smooth visualisation of the hands' motion on the HMD. Whereas the leap-frames transmitted to the server are managed by the Unity HLAPI (Sec. 4.1), which we believe it uses an internal mechanism to regulate the bandwidth in a multi-user experience.

Fig. 5 shows the bandwidth utilised by the Cardboard server in a scenario with three users connected to the same VR environment: one user is the server and the other two are the clients. The bandwidth variation on the server when one client interacts with one hand first and then two hands is shown in Fig. 5a. From this figure we can observe that although the Cardboard downloads leap-frames from the computer at ∼600KB/s, the bandwidth used by the server to download leap-frames from clients is ∼120KB/s
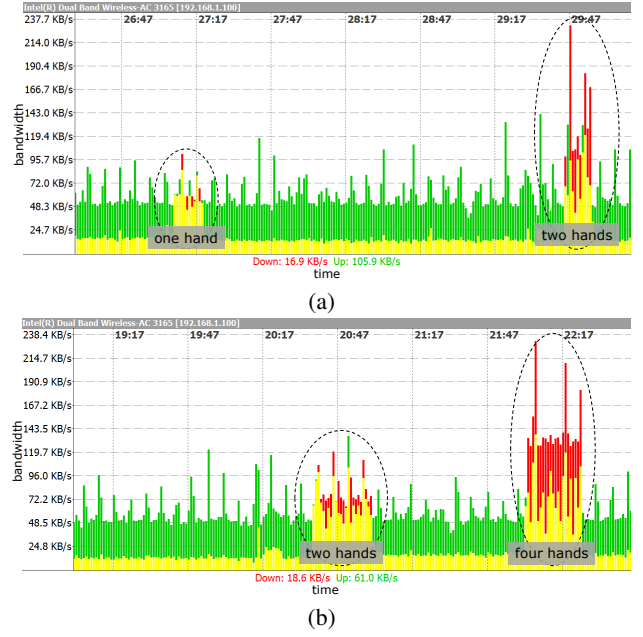


(a)



(b)

Figure 5. Bandwidth that is measured on the Cardboard server in a scenario with three users connected to the same VR environment: one user is the server and the others are the clients. The two graphs show the bandwidth variations on the server (a) when one client interacts with one hand first and then two hands, and (b) when two clients both interact with one hand first and then two hands. Red bars are the bandwidth utilised by the server to download leap-frames from the clients. Green bars are the bandwidth utilised by the server to upload (broadcast) leap-frames to the clients. Yellow bars are used to represent the overlap between the download and upload.

and ∼70KB/s to upload (broadcast) leap-frames to the clients when two hands are used. Moreover, it is interesting to observe that when two clients interact with both their hands the bandwidth usage shown in Fig. 5b is similar to that of Fig. 5a. This bandwidth behaviour is due to the sequential circular counter that regulates the network traffic based on the number of connected clients (Sec. 4.1).

## 5.   VIRTUAL COLLABORATION USING USER-GENERATED MODELS

### 5.1   Model acquisition

We use a smartphone-based 3D reconstruction pipeline to scan real-world objects and create their 3D models that we can import into a virtual environment (Nocerino et al., 2017). A smartphone App acquires images and uploads them to a reconstruction server that progressively creates 3D models using algorithms such as Structure from Motion (Wu, 2013), Multi-View Stereo (Locher et al., 2016) and Meshing (Kazhdan et al., 2006) (Schonberger and Frahm, 2016). While a user is scanning an object, the App receives real-time feedback about the status of the reconstruction from the server, thus enabling them to focus on parts of the scanned object that deserve more attention. Such a client-server architecture produces high-quality 3D models without placing high resource demands on the smartphone. The created 3D models are then stored on a reconstruction server and can be downloaded via a RESTful API (RESTful APIs, 2017). We use these user-generated models to populate the VR space that users collaborate in.
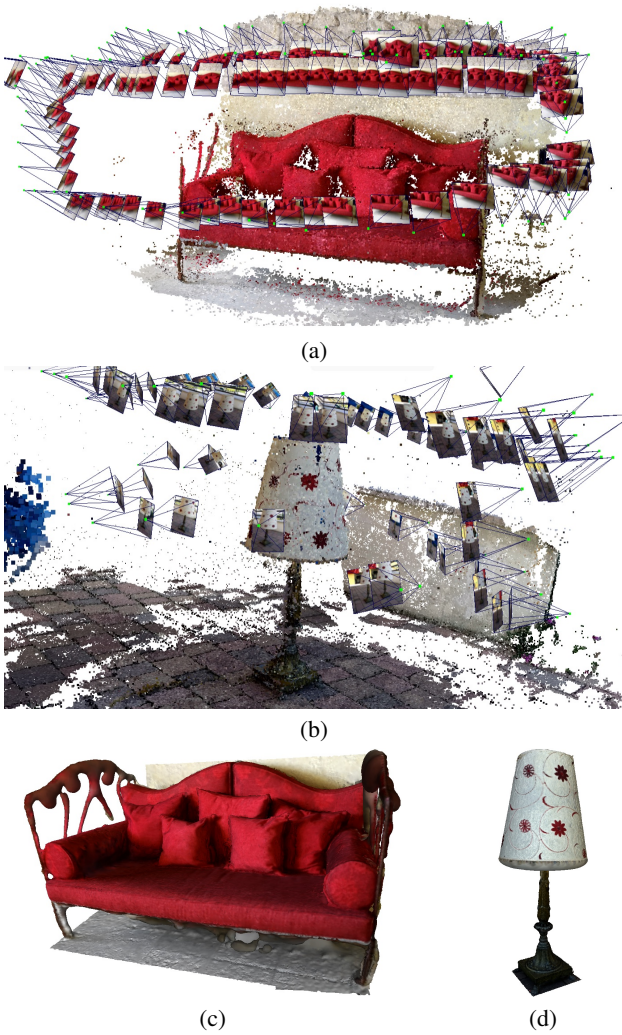
(a)



(b)



(c)        (d)

Figure 6. User-generated 3D models using a smartphone-based reconstruction pipeline. Dense point clouds of (a) a couch and (b) a lamp. Oriented images used for the reconstruction can be observed around the dense point clouds. (c,d) 3D models after meshing operation.

Fig. 6 shows two examples of dense point clouds (i.e. a couch and a lamp) obtained with this reconstruction pipeline. Oriented images used for the reconstruction are displayed around the dense point clouds. We use these dense point to generate the meshed 3D models using Poisson reconstruction (Fig. 6) (Kazhdan et al., 2006). We will include these 3D models in our VR environment to simulate a crime scenario.

### 5.2 Collaborative virtual environment: forensic use-case

The use-case shown in Fig. 7 involves three users concurrently connected via WiFi in the same local network where two of these were connected through their personal Leap Motion to interact via gestures. Users can move their character using the joypad that is connected to the Cardboard via Bluetooth. We authored the VR environment using the 3D models obtained with the pipeline described in Sec. 5.1 and included a reconstructed couch, a re-constructed lamp and a knife. The lamp and the knife have been positioned lying on the floor to simulate a crime scene. Fig. 7a shows one user's point of view that observes the green user who is interacting with his hands. In Fig. 7b both the black and green user are interacting with their hands, and the hands are visible in each other's views. Fig. 7a and b show the VR view in 2D for
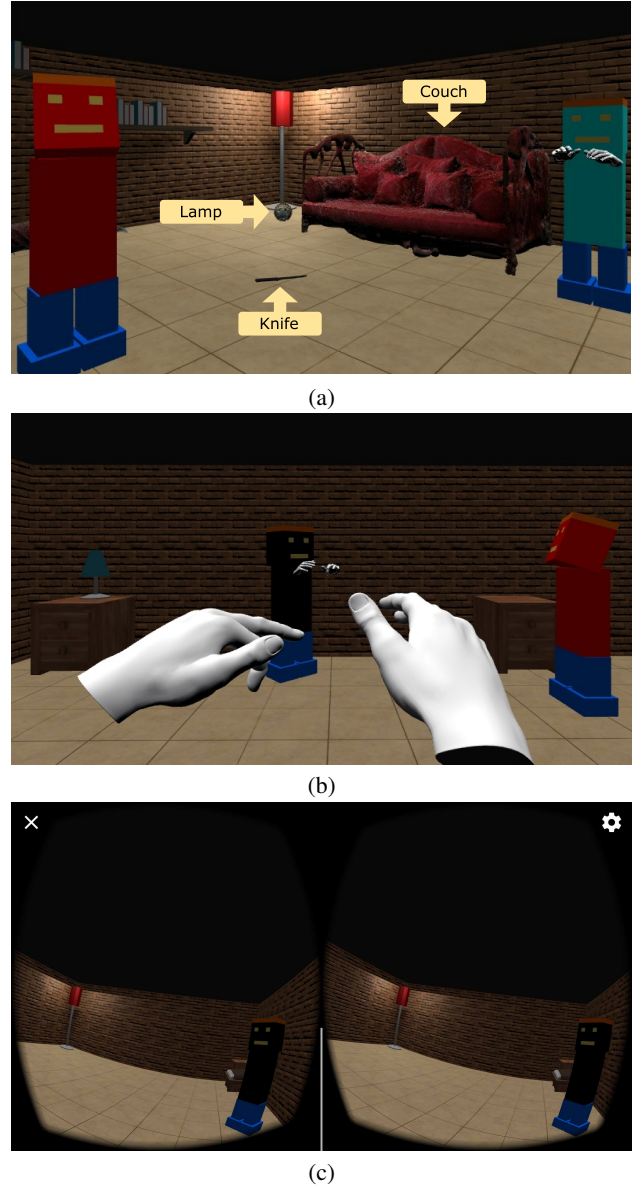


(a)



(b)



(c)

Figure 7. Collaborative virtual environment associated to a forensic use-case. (a) The VR environment includes a reconstructed couch, a reconstructed lamp and a knife. The lamp and the knife have been positioned lying on the floor to simulate a crime scene. (b) Users can interact and see each other's gestures. (a,b) 2D visualisations of users' points of view, whereas (c) Cardboard's point of view while a user is observing another user.

convenience and better representation on paper. In Fig. 7c we included an actual view of the red user trough Cardboard visualisation while looking towards the black user. Because the Cardboard can track the head motion of the user that is wearing it, head rotations can also be visualised and updated in the collaborative VR environment. In Fig. 7b we can see that the red character is looking at the black one with its head tilted. In this way it is possible for users to understand where everyone is looking at.

### 6. CONCLUSIONS

We presented an immersive VR setup that enables collaborations in virtual environments where users can interact using gesture-based commands. Users can both move and see each other's actions in the VR environment, therefore enabling shared visualisations

and collective thinking. Our VR setup uses Google Cardboard to process and display VR, and Leap Motion to track hands and fingers. Cardboard devices communicate through WiFi thus facilitating users to collaborate in the same VR environment without necessary being in the same physical place. We have showed the usefulness of our system through a forensic use-case, where objects of a simulated crime scene were reconstructed using a smartphone-based reconstruction pipeline (Nocerino et al., 2017) and then included in the VR environment.

We are improving this VR setup by reducing interaction delays, visualisations and the authoring of VR environments. We will perform in-depth analysis about scalability of the collaborative VR setup. In order to make the VR application more user-friendly, we will create an interface that will help users to author a VR environment with simple tools. We are also working on the integration of audio communications amongst participants to make the VR experience more immersive. Moreover, we will integrate the module for hand tracking directly in the Cardboard device, as soon as the Leap Motion SDK permits processing on-board a smartphone. This will create a fully integrated VR system for collaborative virtual environments running on everyone's mobile device.

## ACKNOWLEDGEMENTS

## REFERENCES

Abari, O., Bharadia, D., Duffield, A. and Katabi, D., 2016. Cutting the cord in virtual reality. In: *Workshop on Hot Topics in Networks*, Atlanta, US.

Burton, A., Schofield, D. and Goodwin, L., 2005. Gates of global perception: forensic graphics for evidence presentation. In: *Proc. of ACM Multimedia*, Singapore.

Butcher, P., Roberts, J. and Ritsos, P., 2016. Immersive analytics with webvr and google cardboard. In: *Proc. of IEEE Visualization*, Baltimore, US.

Carrozza, L., Tingdahl, D., Bosche, F. and Van Gool, L., 2014. Markerless vision-based augmented reality for urban planning. *Computer-Aided Civil and Infrastructure Engineering* 29(1), pp. 2–17.

Carrozzino, M. and Bergamasco, M., 2010. Beyond virtual museums: Experiencing immersive virtual reality in real museums. *Journal of Cultural Heritage* 11(4), pp. 452–458.

Castelvecchi, D., 2016. Low-cost headsets boost virtual reality's lab appeal. *Nature* 533(7602), pp. 153–154.

Churchill, E. and Snowdon, D., 1998. Collaborative virtual environments: An introductory review of issues and systems. *Virtual Reality* 3(1), pp. 3–15.

Ebert, L., Nguyen, T., Breitbeck, R., Braun, M., Thali, M. and Ross, S., 2014. The forensic holodeck: an immersive display for forensic crime scene reconstructions. *Forensic Science, Medicine, and Pathology* 10(4), pp. 623–626.

EDUROAM network, 2017. `www.eduroam.org`. Accessed: Oct. 2017.

Greenwald, S., Corning, W. and Maes, P., 2017a. Multi-user framework for collaboration and co-creation in virtual reality. In: *Proc. of Conference on Computer Supported Collaborative Learning*, Philadelphia, US.

Greenwald, S., Kulik, A., Kunert, A. and et al., 2017b. Technology and applications for collaborative learning in virtual reality. In: *Proc. of Conference on Computer Supported Collaborative Learning*, Philadelphia, US.

High Level API, 2017. `https://docs.unity3d.com/Manual/UNetUsingHLAPI.html`. Accessed: Oct. 2017.

Kazhdan, M., Bolitho, M. and Hoppe, H., 2006. Poisson surface reconstruction. In: *Proc. of Eurographics Symposium on Geometry Processing*, Cagliari, IT.

Kozhevnikov, M. and Gurlitt, J., 2013. Immersive and non-immersive virtual reality system to learn relative motion concepts. In: *Proc. of IEEE Interdisciplinary Engineering Design Education*, Santa Clara, US.

Leap-frame, 2017. `https://developer.leapmotion.com/documentation/java/devguide/Leap_Frames.html`. Accessed: Oct. 2017.

Locher, A., Perdoch, M. and Gool, L. V., 2016. Progressive prioritized Multi-View Stereo. In: *Proc. of Computer Vision and Pattern Recognition*, Las Vegas, US.

Ma, M., Zheng, H. and Lallie, H., 2010. Virtual reality and 3d animation in forensic visualization. *Digital & Multimedia Sciences* 55(5), pp. 1227–1231.

Nocerino, E., Lago, F., Morabito, D., Remondino, F. and et al., 2017. A smartphone-based pipeline for the creative industry - The REPLICATE project. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Nafplio, GR.

Olshannikova, E., Ometov, A., Koucheryavy, Y. and Olsson, T., 2015. Visualizing Big Data with Augmented and Virtual Reality: challenges and research agenda. *Journal of Big Data* 2(22), pp. 1–27.

RESTful APIs, 2017. `www.restapitutorial.com`. Accessed: Oct. 2017.

Schonberger, J. and Frahm, J.-M., 2016. Structure-from-motion revisited. In: *Proc. of Computer Vision and Pattern Recognition*, Las Vegas, US.

Spicer, R., Anglin, J., Krum, D. and Liew, S.-L., 2017. REINVENT: A low-cost, virtual reality brain-computer interface for severe stroke upper limb motor recovery. In: *Proc. of IEEE Virtual Reality*, Los Angeles, US.

Unity 3D, 2017. `https://unity3d.com`. Accessed: Oct. 2017.

Vorlander, M., Schroder, D., Pelzer, S. and Wefers, F., 2015. Virtual reality for architectural acoustics. *Journal of Building Performance Simulation* 8(1), pp. 15–25.

Wu, C., 2013. Towards linear-time incremental Structure From Motion. In: *Proc. of 3D Vision*, Seattle, US.