Augmented Reality Experience with Physical Artefacts

Gianpaolo Palma, Paolo Cignoni
Visual Computing Lab, Istituto di Scienza di Tecnologie dell'Informazione "A. Faedo", CNR

Abstract
This technical report presents a system to improve the engagement of the user in a virtual reality experience using economic, physical copies of real artefacts, made with cheap 3D fabrication technologies. Based on a combination of hardware and software components, the proposed system gives the user the possibility to interact with the physical replica in the virtual environment and to see the appearance of the original artefact. In this way, we overcome the current limitation of the cheap 3D fabrication technologies: a faithful appearance reproduction. Moreover using a consumer device for the real-time hand tracking and a custom electronic controller for the capacitive touch sensing, the system permits the creation of virtual experiences where the user with his hand can change the virtual appearance of the object using a set of personalization actions selectable from a physical 3D printed palette.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF ABBREVIATIONS

FDM: Fuse Deposition Modelling 3D printing technology
HMD: Head Mounted Display

1 Introduction

This technical report describes a system to improve the engagement of the user in a virtual reality experience using physical copies of 3D scanned real artefacts. The physical copies are made with innovative low-cost 3D fabrication technologies, which allow the fast and inexpensive production of multiple copies. The proposed virtual reality system is based on a combination of off the shelf hardware components and custom electronic circuitry connected by specially developed software components. The proposed setup gives the user the possibility to interact in a new way with the virtual environment and the physical replica. The approach allows seeing in a more faithful way the appearance of the original artefact, overcoming the current limitation of the low-cost 3D fabrication technologies. Moreover, by using a consumer device for real-time hand tracking, and a custom electronic controller for the capacitive touch sensing, the system permits the creation of virtual experiences where the user, with his/her hand, can change the virtual appearance of the object using a set of personalization actions selectable from a physical 3D printed palette.

The most interesting result is the feasibility of implementing an innovative touch-enabled interface using off the shelf components and low-cost custom electronics. We have shown that the proposed system offers the users a new creative experience that allows appealingly interacting with physical artefacts while seeing their appearance correctly.

Section 2 is meant as an introduction to the document. Section 3 gives an overview of the proposed system. Section 4 discusses the hardware setup employed by the project, describing in detail the components used, respectively for the physical object tracking, the hand tracking, and the object touch detection. Section 5 describes the software components of the system. Section 6 presents how the implemented system works in practice describing the user experience offered. Section 7 concludes the report, commenting on the advantage of the system and its limitations.

2 Tangible user interface for virtual reality experiences

The emotional impact of real physical objects in relation with virtual representations plays a significant role in the user experience, particularly in the Cultural Heritage field.

However, a number of practical constraints severely limit the wide adoption of replicas and physical artefacts in the context of virtual and physical exhibition visits. The main issue in this context is the problem of the quality/price ratio of the physical reproductions. Current fabrication (especially consumer 3D printing) technologies are quite expensive, and the fabricated artefacts are of a quality that is often perceived as below the threshold of the expected standards for museums. Indeed, the high-quality appearance reproduction of an artefact is a very complicated and costly process that, for practical and economic reasons, cannot be used in this context.

To face these issues, we explore the coupling of virtual reality consumer technologies, with the direct manipulation of fabricated artefacts to use as a tangible user interface in the virtual environment. The main objective is to build a low-cost, virtual reality system that can offer the user the possibility of a mixed virtual/real experience that can be perceived as more accurate. We designed a system where, while interacting with a low-cost physical copy, whose appearance and finishing are far from the original one, a Head Mounted Display (HMD) provides the user an enhanced visual experience. In the system, the feeling of touch over the replica, combined with the high-quality visuals of the HDM, greatly enhances the immersion and emotional impact.

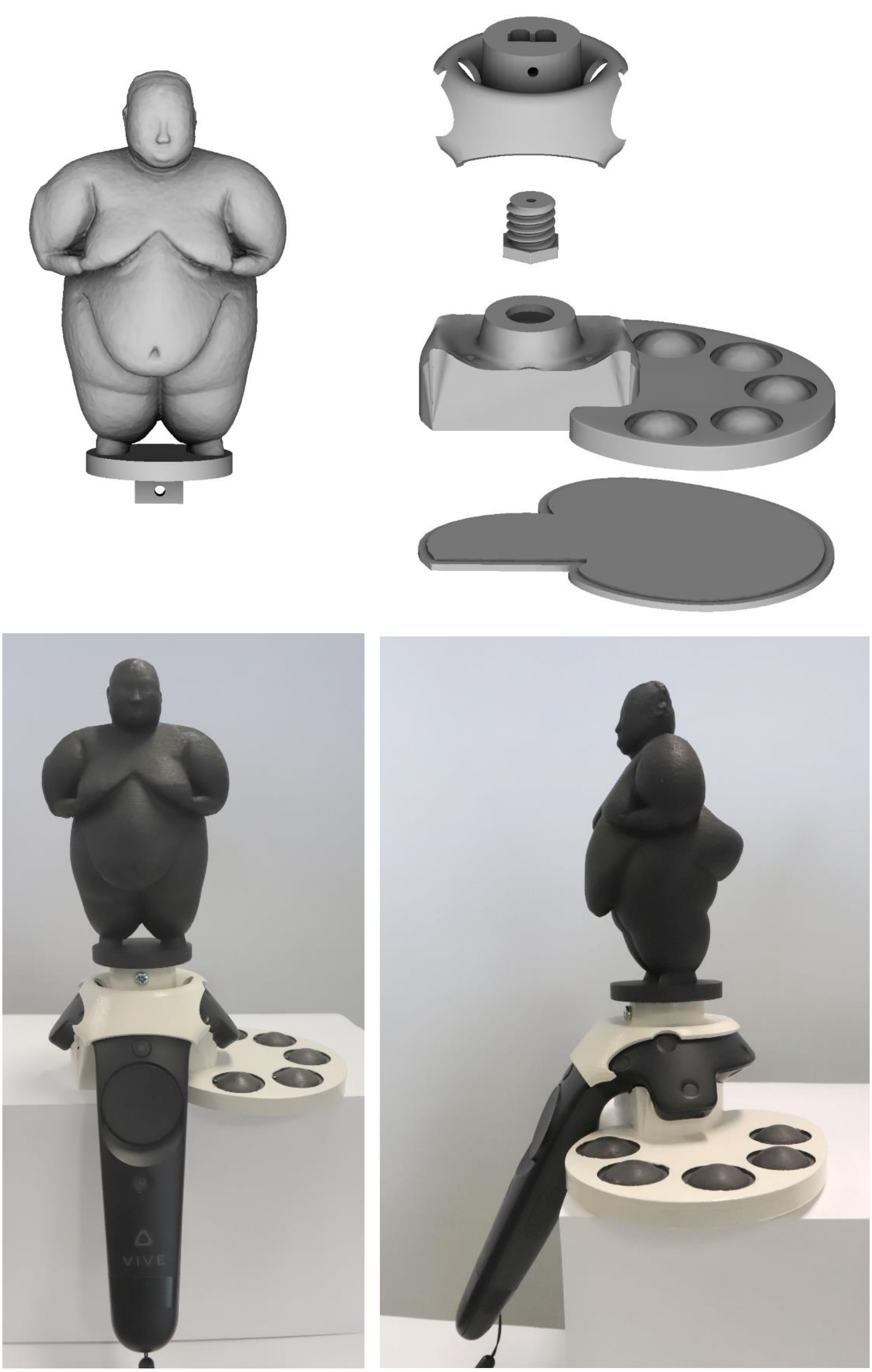

Figure 1: *(Top) CAD models of the parts in the support to mount the physical replica over the HTC Vive input controller. (Bottom) Photos of the 3D printed support with the replica of the Female Figurine mounted over the HTC Vive input controller.*

3 System overview

The main idea of the proposed system is to allow the end user to interact and manipulate the economic physical copy of the artefact in a virtual environment. In this virtual environment, the user can see the correct appearance of the original object and apply a set of personalization actions to change this appearance. The three main goals to reach are:

1. To enhance the visual appearance of an economic physical copy of an artefact using a virtual reality device, i.e., an HMD, to overlay virtually over it the correct appearance of the original object (Figure 4).
2. To enhance the emotional impact of the user giving the possibility to manipulate in his hand the physical replica in the virtual environment, taking advantage of the touch feedback (Figure 6 and Figure 7).
3. To enhance the immersion and engagement of the user, allowing the virtual personalization of the physical replica by changing its virtual appearance when touching its physical surface, using a physical personalization palette (Figure 6).

These goals present the following requirements to satisfy:

1. To compute the position and orientation of the physical object in the virtual environment using a robust tracking solution to guarantee an accurate and precise overlay of the virtual model of the original object over the replica.
2. To show the user's hands in the virtual environment, tracking the movement of each part of both hands to create as realistic as possible virtual hand models.
3. To detect where and when the user touches the replica in order to modify the virtual model in the right position and moment.
4. To satisfy all the previous requirements in real-time to guarantee a high-quality user experience.

To meet all requirements, we designed a system composed of a custom hardware component, described in Section 4, and a software library, described in Section 5. The hardware component is in charge of the tracking of the replica in the virtual environment, the tracking of the user's hands and the detection of the touch event over the surfaces of the replica. The software library, distributed on the replica hardware device and the PC that runs the experience, gathers all the data produced by the hardware devices and returns them visually to the users. This data includes the tracking information of the user in the play area, the tracking data of the hands and the touch event over the replica and the personalization palette.

4 Hardware setup

The proposed hardware component is composed of five parts:

1. the HMD (Figure 2);
2. an active hand tracking device (Figure 2);
3. a 3D printed support to mount the physical replica on the tracking device of the HMD (Figure 1);
4. a physical palette attached to the support to give the user the possibility to select the type of personalization to apply on the surface of the virtual object (Figure 1);
5. an electronic controller to detect the touch over the replica and the personalization palette (Figure3).

The main tasks of the device are the tracking of the physical replica in the virtual environment, described in Section 4.1, the tracking of the user's hands, described in Section 4.2, and the detection of the touch event over the surfaces of the replica, described in Section 4.3.

4.1 Physical object tracking

The device has been designed around the HMD HTC Vive (HTC, 2019), a commercial virtual reality headset composed by a head-mounted helmet, two or more input controllers, and two base stations that use an active optical tracking system to compute the position and orientation of helmet and controllers in the play area with a millimetre accuracy.

The first requirement of our system is to track the position and orientation of the physical replica in the play area in real-time with high accuracy in order to create the relative virtual object in the virtual environment in the right place. To reach this requirement, we mount the physical replica over an HTC Vive input controller rigidly. In this way, the position and the orientation of the replica are described by a simple rigid transformation with respect to the controller. We take advantage of the tracking information of the controller extracted by the HTC Vive driver to compute the actual position of the replica in the play area. The driver extracts this tracking data with a refresh rate of 250Hz.

To mount the replica over the controller, we designed a custom support made using a consumer 3D printer with Fuse Deposition Modelling (FDM) technology. It is built of several pieces modelled to fit the shape of the controller (Figure 1 Top). The top part is modelled to cover as less as possible the controller and its sensors used by the HTC Vive drivers to compute its position and orientation in the play area. In the upper part, it presents a female joint to mount the 3D printed replica without ambiguity and a hole to fasten the replica with a metal screw. The physical replica presents the relative male joint to be fastened on the support. The bottom part is modelled to fit the lower surface of the controller. It is caved to facilitate the fastening with the top part and to host the electronic hardware to detect the touch events. A bolt fastens the bottom and upper part at the controller. On the side of the bottom part, there is the physical palette to select the personalization action to do over the virtual object. Finally, a cap closes the bottom part. Figure 1 shows the CAD models of the pieces of the support and a photo of the mounted device.

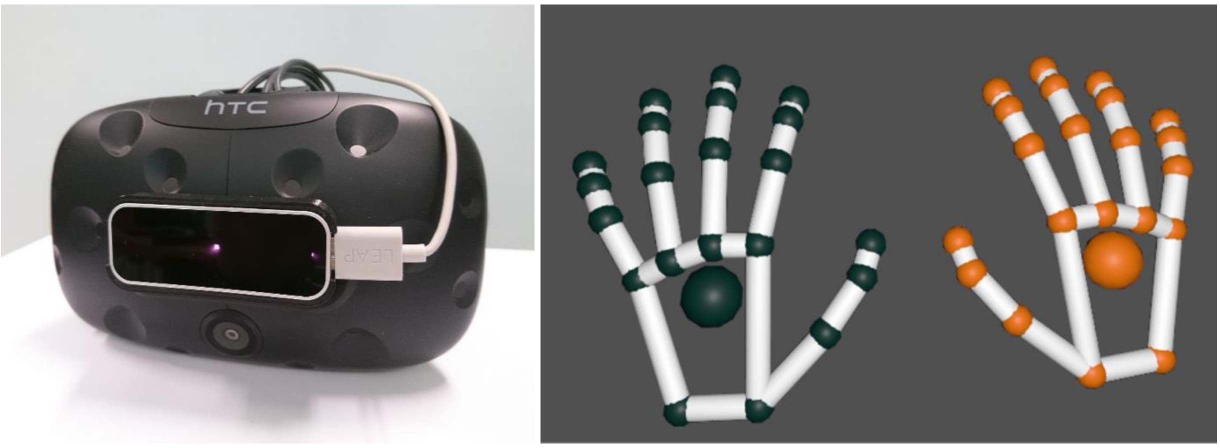

Figure 2: *(Left) Leap Motion device mounted in front of the headset of the HTC Vive. (Right) Stylized model of the hands tracked by the Leap Motion.*

4.2 Hand tracking

The tracking of the hand’s parts is fundamental to give the user the right visual feedback in the virtual environment during the manipulation and interaction with the physical object. To visualize the user’s hands in a correct way and with accurate animations can increase the emotional engagement of the end user in the experiences.

For the hand tracking, we used the specialized sensor device Leap Motion (LeapMotion, 2019) mounted in front of the HTC Vive helmet (Figure 2) rigidly. This sensor allows real-time and accurate tracking of both hands. It returns the position and orientation of all the joints and bones used to model the hand.

The refresh rate of the tracking data is 200Hz. The accuracy of the tracking data is below one centimeter. The calibration of the space of the Leap Motion device with respect to the play area defined by the HTC Vive is provided directly by the Leap Motion. This calibration allows us to know the position and orientation of each hand's part in the coordinate space defined by the play area. Figure 2 (right) shows a stylized model of the hands tracked by the Leap Motion devices in the VR environment.

4.3 Touch detection

The third requirement of the proposed system is to detect in real-time where and when the user touches the replica during the personalization action of its virtual appearance. The main idea is to have a device that can detect the right position touched by the user on the surface of the virtual object and the right moment to apply the personalization. In the last years, several solutions were proposed to track the hand with active 3D depth sensors but they cannot be used for our task because they require huge computational resources, and the tracking accuracy highly degrades when the user manipulates an object in his hands.

For this purpose, we designed a hybrid hardware-software solution. The touch event on the surface (when the user touches the surface) is detected with a custom electronic controller based on the principle of the capacitive touch sensor. The position on the surface of the touch event is computed by a software component using the ray casting procedure explained in Section 2.3.

The electronic controller is based on a low-cost open-source Arduino-like microcontroller with WiFi (NodeCMU, 2019) connected at a shield specialized for the capacitive touch sensing (NXP, 2013) with 12 different input channels. The input channels of this shield are connected to the replica and the button areas of the personalization palette (the five spherical caps on the top of the palette in Figure 1). The buttons on the palette allow the selection of the different personalization actions. Since the replica and the personalization palette are printed with plastic material (PLA) using a classical FDM 3D printer, they cannot be used directly as capacitive sensors because PLA is not conductive. To solve the problem, we spray painted the surface of this object with a Graphite based coating that provides an electricity conductive layer (black surfaces in Figure 1 and Figure 4). The controller is completed with a shield to connect and charge a LIPO battery that powers the device. All the hardware is mounted on a 3D printed support to keep the controller as compact as possible and to facilitate the hosting in the bottom part of the hardware component (Figure 3). The controller simply communicates the state changes (touch or release) of the different input touch channels at the software component via WiFi. The refresh rate of the touch state is 100Hz.

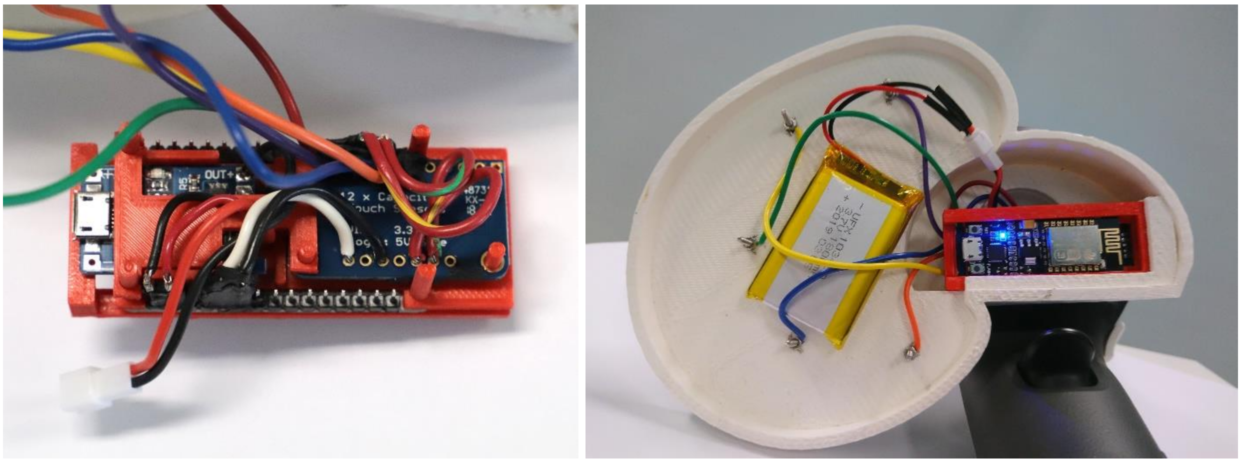

***Figure 3**: (Left) Custom electronic controller to detect the touch over the replica with capacitive touch sensing. (Right) Custom electronic controlled mounted inside the bottom part of the support for the HTC Vive input controller.*

5 Software component

The software component is in charge to gather all the information from the hardware device and to return this data to the end user in a visual way. In particular, it receives the information about position and orientation of the end user's head (the helmet) and of the input controllers from the HTC Vive, the tracking data of the hand from the Leap Motion, and the touch status over the replica and the personalization palette from the electronic controller described in Section 4.3. The head data is used to set the main camera in the virtual experience. The controller tracking data is used for the rendering of the virtual model with the original appearance of the physical replica mounted over it in the virtual environment (Figure 4). The touch status is used during the virtual personalization of the replica.

The software was developed inside the real-time game engine Unity (Unity Technologies, 2019). For the head and controller data, we use the prefab node provider by the SteamVR and LeapMotion. For the management of the touch status, we implemented a custom script that manages the WiFi connection with the electronic controller, analyses the received touch status and translates this status in a visual action. In particular, for the communication with the electronic controller, we use a client-server architecture where the server runs on the electronic controller, and the client runs in the Unity application inside an asynchronous thread. The server waits for an input connection from a client. When this connection is open (at the start-up of the Unity application), the server starts sending messages when there is a change of the touch status over the replica or the palette buttons. Each message contains the id of the touched/untouched area and the touch status (touched/released). The client stores locally the touch status. At each frame drawing, the custom Unity script queries the status inside the client thread and according to the id of the area and the status discriminates the action to do: to apply the personalization on the replica or to change the personalization action. In the case of a touch event over the surface of the replica, the script must compute the position on the virtual object where to change the appearance. For this task, the script integrates all the data coming from the different hardware devices. In particular, it uses a simple ray casting procedure of a set of rays against the virtual model of the physical replica after the transformation tracked by the HTC Vive input controller. Each ray starts from the position of the index finger of the right-hand returned by the Leap Motion. The set of rays is divided into two groups, each one defined by five rays in a cone of directions with aperture 10 degrees. The two main directions of the cones are the direction that connects the index finger position and the current position of the end user's head (the position of the helmet) and the opposite one. We use the opposite directions to be robust concerning the accuracy error (one centimeter) in the hand tracking that sometimes can return a position of the finger slightly inside the virtual model of the replica. This issue is due to the use of different devices with different accuracy for the tracking of the replica and the tracking of the hands. Figure 5 shows a case where in reality the hand is touching the replica but in the virtual world, due the error tracking of the Leap Motion, the virtual hand is not touching the virtual 3D model. If during the ray tracing more than one ray returns a hit with the surface of the virtual model of the replica, we select the intersection closer to the index finger (for the first hit of the series) or closer to the previous intersection (for the other ones). Since the ray casting procedure can be computationally expensive, it could be a bottleneck for the real-time performance requirement of our system. To avoid this problem, we do not use the original 3D model of the object but a simplified version with fewer triangles (1000 triangle) obtained by a quadric edge collapse decimation algorithm (Garland & Heckbert, 1997). This decimation reduces the number of triangles in a controlled way to preserve all the main geometric features introducing a small approximation error.

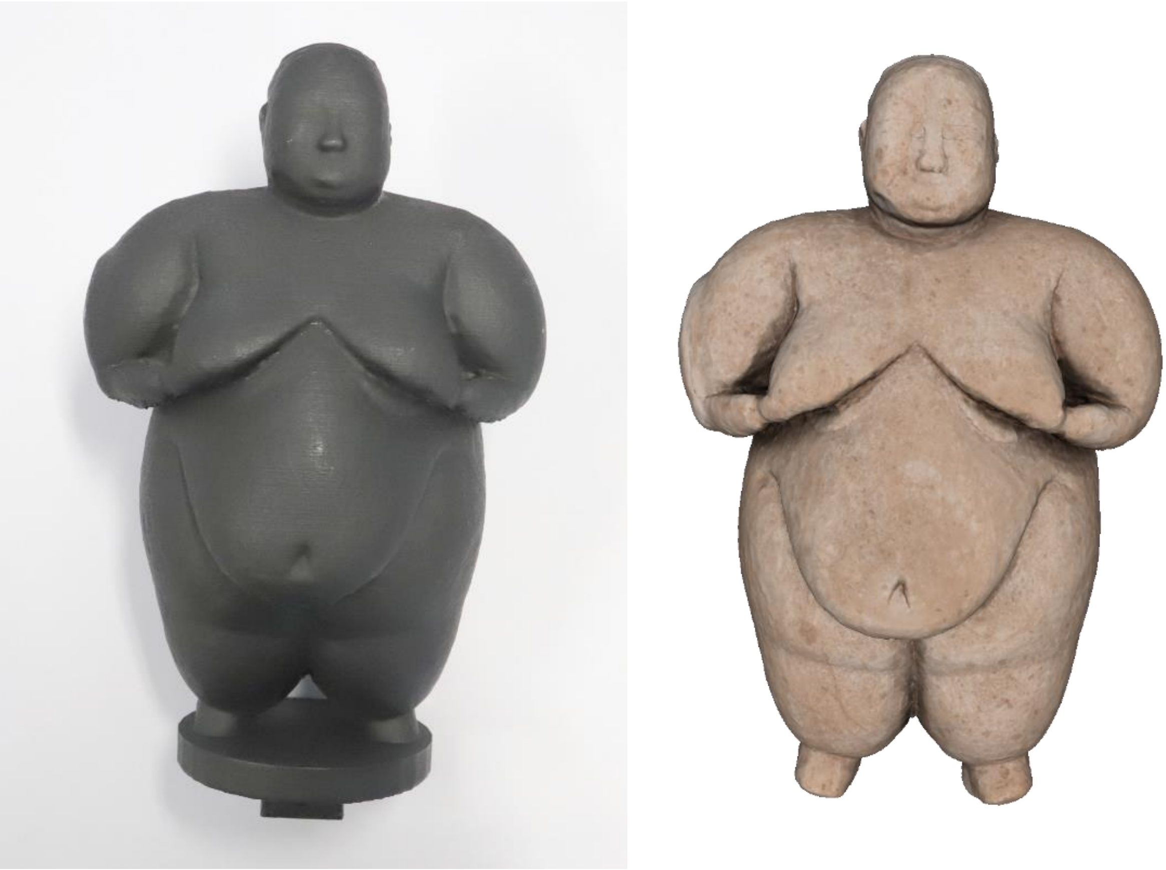

Figure 4*: (Left) Photos of the replica of the Female Figurine obtained with FDM 3D printing technologies and spray-painted with a Graphene conductive coating. (Right) Visualization of the Female Figurine inside the virtual environment with its original appearance.*

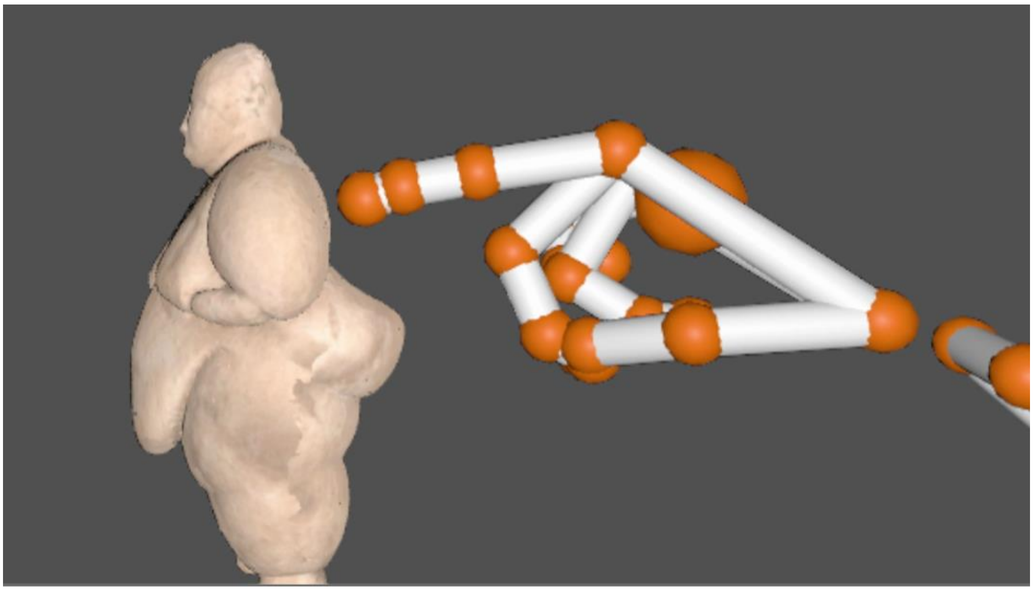

Figure 5*: Calibration misalignment due to the different accuracy of the devices used for the tracking of the replica (HTC Vive) and of the hands (Leap Motion). In this case, the finger is touching the replica in the real world while in the virtual world is not this the case.*

6 User experience

Using the developing system, we designed a virtual experience where the end user cannot only see the original object appearance of the physical replica but can also change this appearance by touching its surface. In particular using the personalization palette, the user can paint over the surface of the replica virtually, using the index finger as a brush, or can attach some additional virtual objects over the surface. By touching on the palette buttons, in the first case, the user selects the color to use for painting, while in the second one the user selects the kind of object to attach. The object is showed over the relative palette button. After the selection on the palette, when the user touches the replica, the system draws on the surface of the virtual object using the paint color or it places the select object in the touch point using also the normal vector to orientate it in a coherent way. After few user testes, the general impression is that the proposed system can engage more the user in the experience due to the touch feedback of the physical replica during the personalization.

Figure 6 shows two frames of a personalization section of the Female figurine, where the user can select four different colors and one object (in this case an hat).

Figure 7 shows some photos of a user during a virtual personalization session of the replica.

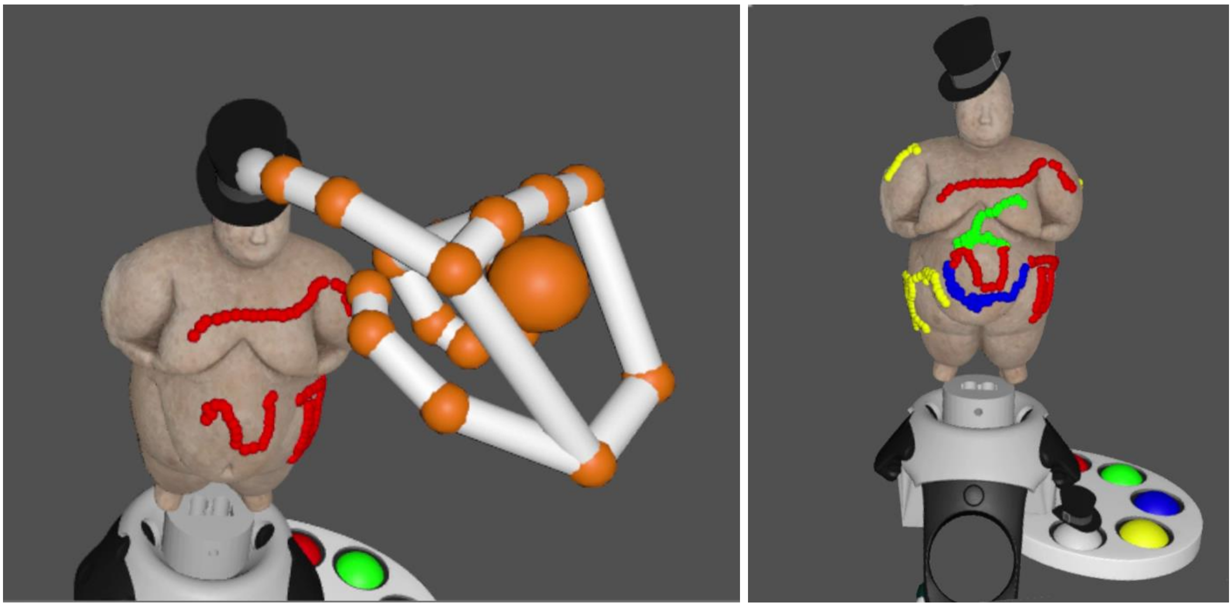

Figure 6*: (Left) Personalization phase of the Female Figurine. (Right) Personalization results at the end of the session.*

7 Conclusion

The designed system is a combination of hardware and software components that permit the user to interact and manipulate the economic physical copy of an artefact in a virtual environment using this physical replica as a tangible user interface. With the proposed system, the end user can see the correct appearance of the original object virtually projected on the physical replica, and it can apply a set of personalization actions to change this appearance. Thanks to the integration of different hardware devices (HTC Vive, Leap Motion, and custom electronic controller) the system can detect the touch events of the user over the replica and compute the touched position in the software component. This position is used to modify the virtual appearance of the replica on the right surface point. Then the design of a personalization palette allows the user the selection of the personalization action to do over the virtual appearance of the object. The system described in the deliverable is built upon collaborative work within the Emotive EU project Consortium, to design, develop and evaluate a variety of use cases within the cultural sites in the project, especially for the Çatalhöyük site.

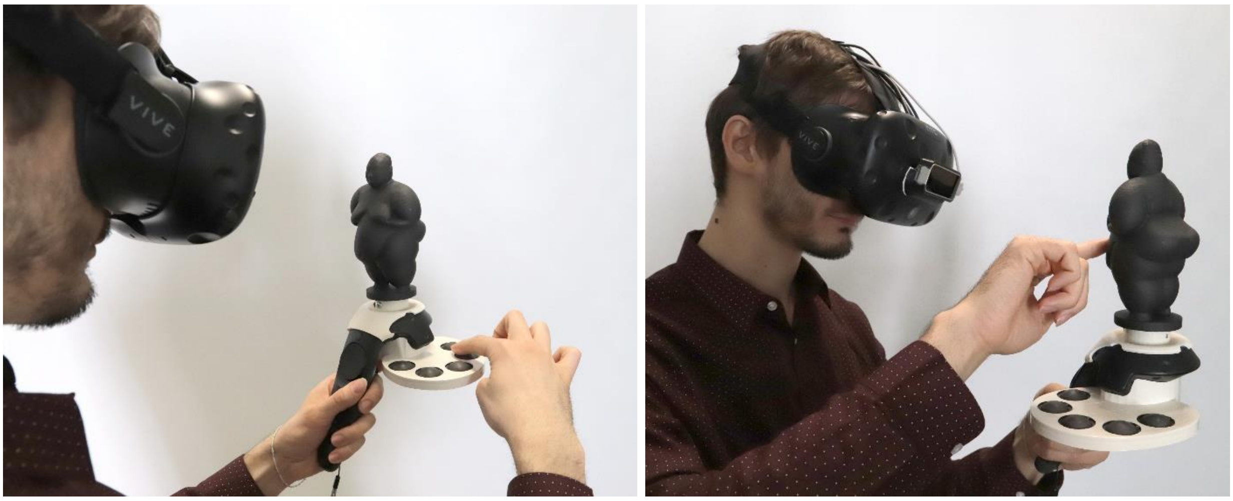

***Figure 7**: Photos of an end user during a replica personalization session.*

8 Bibliography

Garland, M., & Heckbert, P. S. (1997). Surface simplification using quadric error metrics. *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (pp. 209-216). ACM.

HTC. (2019). *HTC Vive Tecnical Specificaation*. Retrieved from https://www.vive.com/us/product/vive-virtual-reality-system/

LeapMotion. (2019). *Leap Motion Techical Specification*. Retrieved from https://www.leapmotion.com/technology/

NodeCMU. (2019). *NodeCMU Technical Specification*. Retrieved from https://www.nodemcu.com/index_en.html

NXP. (2013). *MPR121 Technical Specification*. Retrieved from https://www.nxp.com/docs/en/data-sheet/MPR121.pdf

Unity Technologies. (2019). *Unity Game Engine*. Retrieved from https://unity.com/