Natural interaction with small 3D objects in virtual environments

Irfan Prazina*
Supervised by: dr. Selma Rizvić†

Faculty of Electrical Engineering Sarajevo
University of Sarajevo / Bosnia and Herzegovina

Abstract

This paper presents a novel solution for natural interaction with small 3D objects in virtual environments, using Leap Motion as sensor and WebGL for presenting a virtual hand and 3D objects. Most current solutions for this type of interaction use gestures as a mean of control. The interaction will emulate a natural grab-and-move action of a human hand. The virtual hand will be displayed alongside with a 3D object and will be able to grab and move the object using Cannon.js physics engine. The virtual hand which emulates real hand should minimize the user's learning time.

Keywords: natural interaction, virtual reality, WebGL, user immersion

1 Introduction

Many cultural monuments and artifacts disappear or get destroyed. One way to preserve them for next generations is through technology. Preservation is not enough, as even the best technology if not being used gets forgotten and obsolete. To keep people interested in virtual heritage the new ways of immersive interaction must be developed.

Once we have geometric data of virtual objects and environments, and different ways of presenting them, we must find means of navigation which will help a user to get the most information and best experience.There are different ways of interaction with virtual objects and environments. Users can be interested in a new technology and technique, but that interest will be short lived if it is not easy to use. Our virtual presentations are web based because we would like to present the selected cultural heritage objects to everyone, not just the people who come to visit museums or archaeological sites. Our experience shows that online presentations also attract people to physical locations of monuments.

We will present a way of interaction with virtual objects using only a hand and Leap Motion. This type of interaction will mimic the movements and dynamics of real hand. Users will be able to see a virtual representation of their hand and with very little delay see all the movements they make.

*iprazina1@etf.unsa.ba
†srizvic@etf.unsa.ba

The paper is organized in the following way: Section 2 presents related work, in Section 3 is description of our natural interaction's concept, Section 4 presents a case study and the interaction implementation, Section 5 presents our user evaluation and a conclusion is given in Section 6.

2 Related work

Papers [7], [8] and [5] describe the use of the Leap Motion for a gesture recognition.
In [7] comparison of Kinect and Leap Motion is given. The authors state that Kinect gives more detailed data, but is less accurate, while Leap Motion gives high level data as a set of relevant hand points and pose features.
In [8] Leap Motion is used to recognize sign language gestures. The Sign language gestures recognition can be problematic when one or more fingers are not in line of sight.
In [5] the authors use the Leap Motion API to detect key presses and to create a keyboard like musical instrument.
These works show advantages of Leap Motion over Kinect, like well documented API and user friendly data structure. As drawback Leap Motion tends to lose track of some fingers if line of sight is blocked.
In [4] a detailed analysis of the precision and reliability of Leap Motion is given. The distance from the hand to the sensor is a parameter that significantly affects Leap Motion's consistency and performance. Leap Motion's accuracy significantly drops if the hand is more than 250mm above the controller. Inconsistent sampling frequency is stated as an important limitation of the controller performance.
In [6] Kinect is used to detect human gestures for interaction with 3D objects like moving, opening car doors, selecting a seat, etc.
Common use of Leap Motion is in recognition of gestures and using them as triggers for an action. In our paper hand movements are directly mapped to movement of the virtual hand and the virtual hand interaction with other objects is simulated using physics engine. In [9] the autor describes how a physics engine can be implemented in JavaScript.

3 Our concepts of natural interaction

Natural interaction is often realized with excessive use of gestures and gesture recognition. In this work we tried to make interaction with a virtual object as close to the interaction with a real object as possible. Users need to learn very little before interaction. We achieved this by simulating behavior of a real hand in 3D virtual environment using physics engine for simulating physics and web browser as an execution environment. Web technologies are used because everyone has a web browser, and those who don't have Leap Motion can use a mouse for the interaction with objects.

To simulate the hand behaviour in a virtual environment we used:

- Three.js – JavaScript library for creating and displaying 3D graphics [3], it is based on WebGL,
- Cannon.js – JavaScript library used as physics engine [1] and
- Leap.js – Leap Motion's JavaScript library for acquisition of a Leap Motion data [2].

In this type of interaction physics engine has a pivotal role. If we do not set collisions right, the hand will go through objects or it will push objects before we even get close. Cannon.js is chosen because of its simplicity and speed. Lack of feeling of touch was the problem we anticipated before making this work. When one is interacting in a way described here, grabbing movement can be tiresome, because in some cases one will grab too fast, too strong or too weak and the object will slip. To solve this problem we used Cannon.js lock constraint force. The constraint mimics glue effects, it makes one object stuck to another. If the user puts his palm close enough to an object, the object will stick to the palm and will not slip in case of fast grabbing, very strong or very weak grip. When an object is on a palm the user can rotate it by rotating the palm, as one would rotate it in real life. To make a virtual hand able to hold an object we need to add collision boxes to the hand and to the object. The hand is made of boxes, where one box represents one bone of the hand.

Each finger consists of 4 bones (Figure 1) where the first bone is in the palm and last three are movable finger's bones. Each bone in the hand has a mesh and a body related to it. The mesh is used for 3D presentation, and the body is used for physics simulation. The shape of the mesh is the same as the shape of the body so objects do not go through visual representation of the fingers or avoid them at distance (Figure 2). Leap Motion detects bones positions and orientation data. On each Leap Motion's loop iteration we get new data which are used to move bone bodies; these two steps are done using Cannon.js and Leap.js. After bone body has been moved, the mesh position must be updated; this is done in Cannon.js loop. If some bone

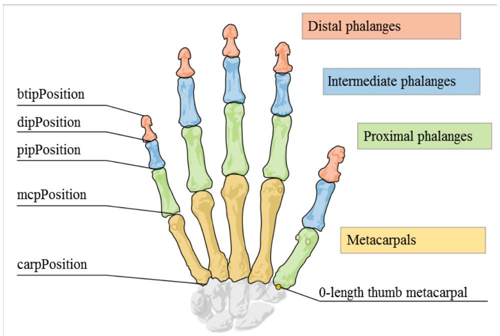

Figure 1: Structure of the virtual hand, picture is from developer.leapmotion.com

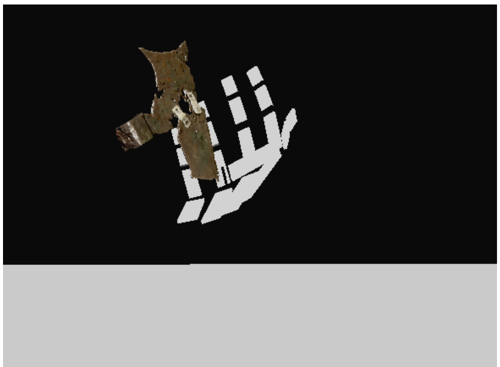

Figure 2: Display of the virtual hand

has hit an object Cannon.js will calculate its velocity and object's position will be updated accordingly. The final step in the iteration of Leap Motion's loop is rendering changes in the scene (Figure 3).
One of the challenges was to synchronize all coordinate systems. Each library has its own coordinate systems for bone positions (Leap.js), body positions (Cannon.js) and mesh positions (Three.js). Details on coordinate systems synchronizations will be explained in Section 4.2.

4 Case study

4.1 The White Bastion project

The fortification known as "White Bastion" is one of the most impressive and important historical sites in Sarajevo. It is located on the south-east outskirts of the city, with an overview of the city valley. Through the history it had a very significant and strategic position. The fortification is a part of the dominant defense walls that were surrounding the old city of "Vratnik". The value of the historical site presents the various strata, starting from medieval until the present time. During the archaeological excavations remains from medieval fortification from 14th century were found, Ottomans period (17th century) when the fortification was expanded and some new objects were built. During Austro-Hungarian rule the part of the fortification and

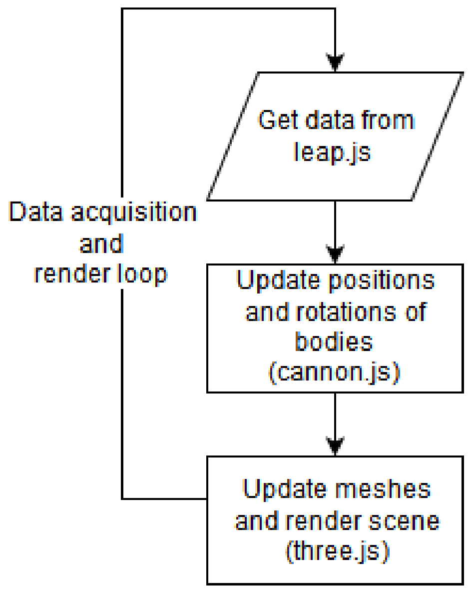

Figure 3: The algorithm diagram

the object inside the walls were demolished and destroyed and a new group of objects was built. During the early excavation, a significant number of artifacts was found, registered and conserved for the purpose of the exhibition hosted in Museum of Sarajevo. 4D Virtual presentation of White Bastion aims to present the historical development of this cultural heritage object through digital stories combined with interactive 3D models of the Bastion in various time periods. These models contain digitized findings from the site and their 3D reconstructions. Structure of the project is displayed in Figure 4.

4.2 Implementation details

The implementation of the idea required the use of three libraries: one for physics simulation, one for display and one for acquiring data. The main challenge of this implementation was to synchronize all the position and orientation data, and all the coordinate systems. Leap Motion uses the three dimensional right-handed Cartesian coordinate system which has millimeters as units. The x axis is placed on the longer side of the device. Leap Motion uses a rotation matrix to represent bone's orientation. Cannon.js will calculate data in any coordinate system, but one coordinate system must be used consistently. Body's orientation is represented in the form of the quaternions. Three.js uses the right-hand coordinate system where z axis is perpendicular to the screen oriented toward a user. Leap Motion coordinate system is used as a reference, mesh and body positions are set accordingly. Leap Motion uses loop iterations to update bone position and orientation. We must update body and mesh positions on each loop iteration. In the first loop iteration we set mesh position directly to be equal to bone positions. This is the initialization phase in which each mesh gets the position of its corresponding bone.

```
mesh
    .position
    .set(
        bone.center()[0],
        bone.center()[1],
        bone.center()[2]
);

mesh.setRotationFromMatrix(
    (new THREE.Matrix4)
        .fromArray(bone.matrix())
);
```

In the following iterations a mesh position is set using the body position. The body position is set in a same way as mesh position with exception of orientation. The orientation of the body is set using the quaternion helper object. Helper object is initialized using bone rotation matrix, and its orientation is adjusted to a base bone rotation. Base bone rotation is set to Euler angles $(\frac{\Pi}{2}, 0, 0)$.

```
quat = new THREE.Quaternion();
quat
    .setFromRotationMatrix(
        (new THREE.Matrix4)
        .fromArray( bone.matrix())
);
quat.multiply(baseBoneRotation);
body.quaternion.set(
    quat.x,
    quat.y,
    quat.z,
    quat.w
);
```

In this way the virtual hand's movement is solved. Setting the positions of meshes according to the positions of the bodies ensures that objects underlie the physics simulation. To solve afore mentioned problems of lack of tactile feedback and problem of "slippery" objects we added lock constraint force which activates when the palm is at a distance less than 50 units. In that case a new constraint object is created between small 3D object and palm bone. The small 3D object will still move inside of the hand and it will appear as it is stuck with one side to the palm so it will not slip away if the grip is fast, too strong or too weak. The constraint is deactivated when user moves away his hand.

There are two events that must be handled: when a hand is placed above the sensor and when a hand leaves sensor's space. On first event bodies and meshes are created and the Leap Motion loop starts updating bone positions. On second event all bodies and meshes must be destroyed.

The described interaction is added to the White Bastion project. In the White Bastion project there is a web page for each of the digitalized archaeological findings. The web page's layout is made of two Three.js scenes; in the first scene 3D model of the excavated object is presented and in the second scene the reconstruction of the object is displayed. In the initial version of the project objects can be viewed from all sides using mouse for movement and rotation. In the new version the Leap Motion interaction is added for the museum setup. An example web page is presented in Figure 5.

5 Evaluation

The Leap Motion interaction is evaluated using qualitative user experience analysis. Participants in the analysis have different educational backgrounds, age and gender. Ten participants have been interviewed. The interview has been structured in such a way that participants can share their impressions on the Leap Motion interaction implementation and the interaction using a computer mouse. Users had enough time to experience both interactions. They were asked to look the object using mouse, and after they explored the object from all sides they were introduced to Leap Motion. In the Leap Motion interaction users were asked to look at same object, and how hard was for them to see all the sides and details using the new interaction. Hypotheses that are being checked in the interview are presented in Table 1. The interview is structured in form of open questions which participants used as guidelines. This type of interview is used to give participants opportunity to express their opinions. Questions are formulated in a way that one question can give information about one or more afore mentioned hypotheses. List of questions is given in Tables 2 and 3. In the interview most of the subjects had similar experience while using the interaction with Leap Motion. All of them spent more time using Leap Motion than the mouse interaction, but they found mouse interaction easier to use so H1 has not been confirmed. Even though participants found mouse easier to use, they also found the Leap Motion interaction more interesting and immersive, which proves H2. Some of them made statements like: "Leap Motion is interesting, has potential, but less precise", "When using the Leap Motion all focus is on the 3D object", "Leap Motion can make people interested in viewing all these objects", etc. (H2) All participants claimed that mouse interaction is more precise and reliable but dull and less immersive. Participants had little experience with virtual cultural heritage and they stated that projects like this make them interested in history and cultural heritage, thereby confirming H3.

H1	Leap Motion interaction is easier to use than mouse
H2	Leap Motion interaction is more engaging
H3	Leap Motion can make people interested in virtual heritage

Table 1: List of hypotheses

Q1	What is your opinion on interaction with a virtual object using mouse? (What are advantages and disadvantages?)
Q2	What is your opinion on interaction with a virtual object using Leap Motion?(What are advantages and disadvantages?)
Q3	What is easier to use and why?

Table 2: Questions for hypotheses H1 and H2

Q4	What is your experience with virtual scenes and virtual cultural heritage?
Q5	What do you think about virtual heritage and this type of projects?

Table 3: Questions for hypothesis H3

6 Conclusion

In this paper we presented a novel solution for natural interaction with small 3D objects in virtual environments, using Leap Motion as sensor and WebGL for presenting virtual hand and 3D objects. In the qualitative user experience analysis of this method the participants noticed that the Leap Motion interaction has precision and reliability issues in comparison with the traditional mouse interaction. However, they spent more time using the Leap Motion interaction, and they found that interaction immersive and interesting. Participants stated that they had more fun and found new experience more engaging. This type of interaction could be viable if precision and speed of the Leap Motion controller is improved. In the current state Leap Motion can still be used as presented because of its immersive potential, but improvements are needed.

References

[1] Cannon.js. http://www.cannonjs.org. Accessed: 2016-02-14.

[2] Leap motion. http://developer.leapmotion.org. Accessed: 2016-02-14.

[3] Three.js. http://threejs.org. Accessed: 2016-02-14.

[4] Jože Guna, Grega Jakus, and Matevž Pogačnik. An analysis of the precision and realiability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors, 2014.

[5] Jihyun Han and Nicolas Gold. Lessons learned in exploring the leap motion sensor for gesture-based instrument design. Internation Conference on New Interfaces for Musical Expression, 2014.

[6] Jong-Oh Kim, Mihye Kim, and Kwan-Hee Yoo. Realtime hand gesture-based interaction with objects in 3d virtual environments. International Journal of Multimedia and Ubiquitous Engineering, 2013.

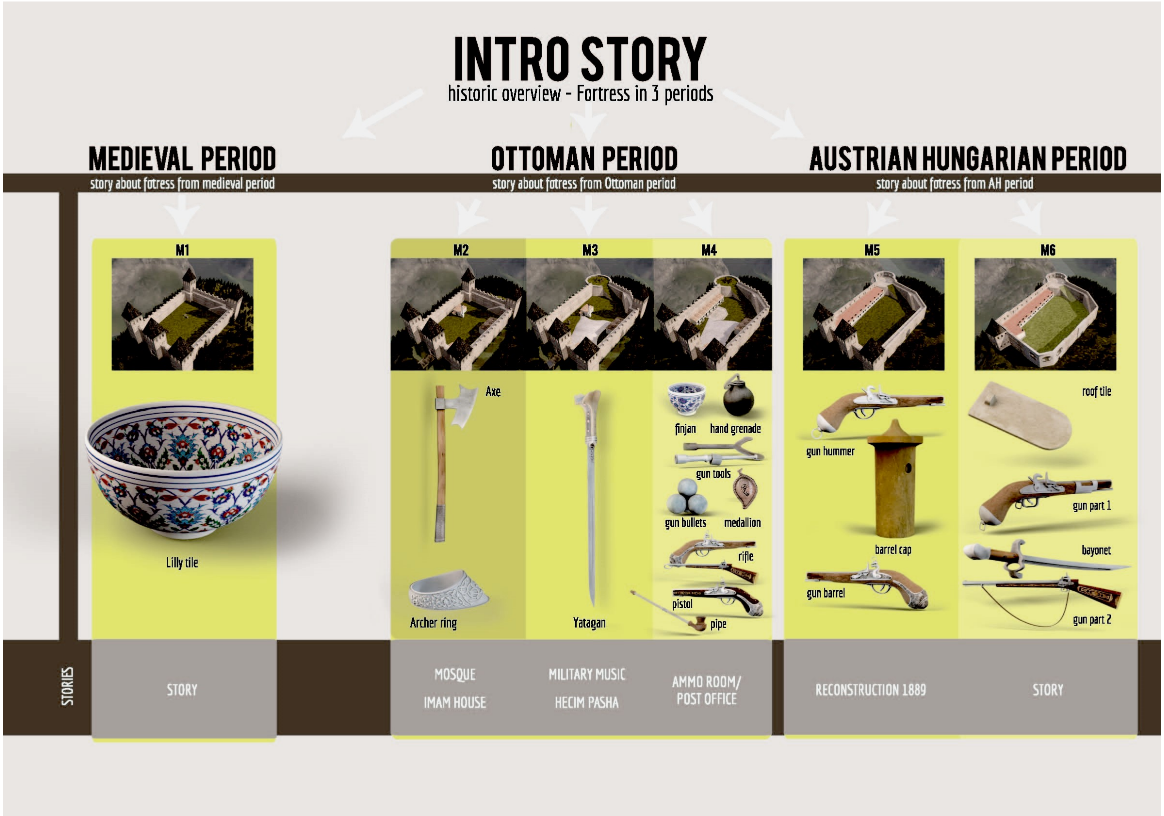

Figure 4: Project structure

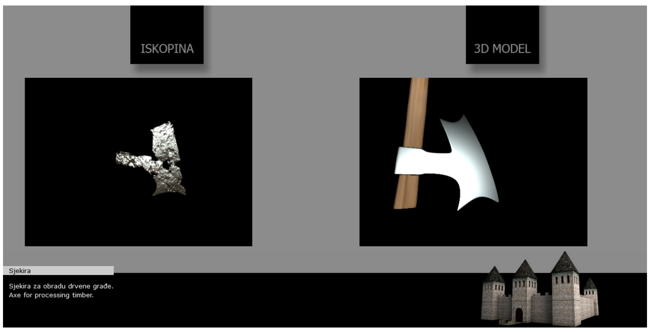

Figure 5: A 3D object web page from White Bastion project

[7] Giulio Martins, Fabio Dominio, and Pietro Zanuttigh. Hand gesture recognition with leap motion and kinect devices. ICIP, 2014.

[8] Leigh Ellen Potter, Jake Araullo, and Lewis Carter. The leap motion controller: A view on sign language. 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, 2013.

[9] Adam Ranfelt. Build a simple 2d physics engine for javascript games. IBM developerWork, 2012.