Real-Time Visualization of Muscle Activity using Augmented Reality and Motion Capture

Nikolai René Berg

Thesis submitted for the degree of Master in Robotics and Intelligent Systems
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2020
Real-Time Visualization of Muscle Activity using Augmented Reality and Motion Capture

Nikolai René Berg
Real-Time Visualization of Muscle Activity using Augmented Reality and Motion Capture

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo
Abstract

Augmented reality sees increased use in existing forms of entertainment to attract new audiences and elevate the experience. In performances like concerts and dance, the use of augmented reality is still heavily experimental, but shows great potential. However, applying augmented reality visual effects to a performance requires additional planning, costs, and time, which makes it unappealing to low budget performances.

This thesis approaches this problem by creating an augmented reality system that dynamically generates visual effects in real-time using muscle activity. Due to this dynamic generation of visual effects, fewer resources are needed to design the visuals, and it also opens up possibilities for improvisation with the effects. Subjects’ perception of muscle activity is also investigated to see if the visualization makes it easier to interpret the performer’s movement and gestures.

The system was created using motion capture, a Myo armband, and the AR platform ARCore, which lets the audience use their personal mobile device to view the visual effects and share the same experience. The augmented reality system was implemented successfully, but not tested with an audience.

To test how subjects perceive muscle intensities, two online user studies were conducted. 37 and 23 subjects participated, respectively. By gathering both quantitative and qualitative data, the subjects’ perception of muscle activities was measured. The results show a significant improvement in the subjects’ perception of muscle intensities.
## Contents

1 Introduction .................................................. 1
   1.1 Thesis goals ............................................. 1
   1.2 Contributions ........................................... 2
   1.3 Thesis structure ........................................ 2

2 Background .................................................. 3
   2.1 Virtual reality ........................................... 3
   2.2 Augmented reality ....................................... 4
       2.2.1 Augmenting the world around you ............... 5
       2.2.2 Applications of augmented reality ............. 6
   2.3 Related Work ............................................ 7
       2.3.1 Augmented reality in dance performances ...... 7
       2.3.2 Performance interaction using mobile devices ... 9
       2.3.3 Real-time human arm tracking ................... 10
   2.4 Statistical method ..................................... 11
       2.4.1 Null hypothesis and alternative hypothesis ... 11
       2.4.2 P-values ........................................... 11
       2.4.3 Chi-square test ................................... 12
       2.4.4 Mann-Whitney U test ............................. 12
       2.4.5 Likert scale ...................................... 12

3 Software and tools .......................................... 13
   3.1 Myo armband ............................................. 13
   3.2 OptiTrack & Motive ..................................... 13
   3.3 Unity .................................................... 14
       3.3.1 Scripting ........................................... 14
       3.3.2 Objects ............................................ 14
       3.3.3 OptiTrack plugin .................................. 16
   3.4 ARCore .................................................. 17
   3.5 Google Cloud Platform .................................. 18

4 Implementation .............................................. 19
   4.1 System development .................................... 19
       4.1.1 Prototype .......................................... 19
       4.1.2 Selecting a tracking method .................... 21
       4.1.3 ARCore & motion capture ....................... 23
       4.1.4 System overview .................................. 29
4.2 Visualization and user experience
  4.2.1 Particle systems
  4.2.2 Effortless connection
  4.2.3 Usability testing
  4.2.4 Interactive elements

4.3 Video demo

4.4 Experiment planning
  4.4.1 Original experiment
  4.4.2 AR only experiment

5 Experiments and results
  5.1 Primary experiment
    5.1.1 The questionnaire
  5.2 Primary experiment results
    5.2.1 Fist gesture
    5.2.2 Fingers spread gesture
    5.2.3 Matching pairs
    5.2.4 The questions
  5.3 Primary experiment analysis
    5.3.1 Fist gesture analysis
    5.3.2 Fingers spread gesture analysis
    5.3.3 Matching pairs analysis
    5.3.4 Question analysis
  5.4 Secondary experiment
    5.4.1 The questionnaire
  5.5 Secondary experiment results
    5.5.1 Sound
    5.5.2 Sound and visuals
  5.6 Secondary experiment analysis
    5.6.1 Video analysis
    5.6.2 Question analysis
    5.6.3 Comparing with the primary experiment

6 Discussion
  6.1 The questionnaires
  6.2 The app
  6.3 Visualization using augmented reality
  6.4 Future work
    6.4.1 Live performance
    6.4.2 Applying the system to a robot
    6.4.3 Further development
  6.5 Conclusion
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Lighthouse tracking</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Virtual chair</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Perspective view in Motive</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>Order of Execution</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Green sphere in Unity</td>
<td>16</td>
</tr>
<tr>
<td>3.4</td>
<td>Example script</td>
<td>17</td>
</tr>
<tr>
<td>3.5</td>
<td>Cloud anchor</td>
<td>18</td>
</tr>
<tr>
<td>4.1</td>
<td>Target cylinder</td>
<td>20</td>
</tr>
<tr>
<td>4.2</td>
<td>Prototype animations</td>
<td>21</td>
</tr>
<tr>
<td>4.3</td>
<td>Running TensorFlow model</td>
<td>23</td>
</tr>
<tr>
<td>4.4</td>
<td>Desired behavior</td>
<td>24</td>
</tr>
<tr>
<td>4.5</td>
<td>Cloud anchor sample</td>
<td>25</td>
</tr>
<tr>
<td>4.6</td>
<td>Matching the coordinate frames</td>
<td>26</td>
</tr>
<tr>
<td>4.7</td>
<td>OptiTrack calibration triangle</td>
<td>27</td>
</tr>
<tr>
<td>4.8</td>
<td>Setup process in the application</td>
<td>27</td>
</tr>
<tr>
<td>4.9</td>
<td>Moving the motion-tracked object</td>
<td>28</td>
</tr>
<tr>
<td>4.10</td>
<td>Drawing in 3D space</td>
<td>28</td>
</tr>
<tr>
<td>4.11</td>
<td>The core system</td>
<td>30</td>
</tr>
<tr>
<td>4.12</td>
<td>Particle system drawing</td>
<td>32</td>
</tr>
<tr>
<td>4.13</td>
<td>Particle system test in action</td>
<td>32</td>
</tr>
<tr>
<td>4.14</td>
<td>Particle system A</td>
<td>33</td>
</tr>
<tr>
<td>4.15</td>
<td>Particle system B</td>
<td>33</td>
</tr>
<tr>
<td>4.16</td>
<td>Combined particle systems</td>
<td>33</td>
</tr>
<tr>
<td>4.17</td>
<td>The defined material used to render particles</td>
<td>34</td>
</tr>
<tr>
<td>4.18</td>
<td>Upgraded armband</td>
<td>34</td>
</tr>
<tr>
<td>4.19</td>
<td>Sliders</td>
<td>36</td>
</tr>
<tr>
<td>4.20</td>
<td>The system in use, shown from three views</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>Questionnaire image 1</td>
<td>40</td>
</tr>
<tr>
<td>5.2</td>
<td>Questionnaire image 2</td>
<td>40</td>
</tr>
<tr>
<td>5.3</td>
<td>Questionnaire image 3</td>
<td>41</td>
</tr>
<tr>
<td>5.4</td>
<td>Questionnaire image 4</td>
<td>41</td>
</tr>
<tr>
<td>5.5</td>
<td>Questionnaire image 5</td>
<td>42</td>
</tr>
<tr>
<td>5.6</td>
<td>Questionnaire video</td>
<td>43</td>
</tr>
<tr>
<td>5.7</td>
<td>Task 1 (Fist close-up) votes</td>
<td>44</td>
</tr>
<tr>
<td>5.8</td>
<td>Task 2 (Fist distant) votes</td>
<td>44</td>
</tr>
</tbody>
</table>
## List of Tables

4.1 Color and Myo channel correspondence .......................... 32
4.2 Parameters of the particle systems ................................. 34

5.1 Task 1-3 votes in order ........................................... 45
5.2 Task 4 & 5 votes in order ........................................... 46
5.3 Task 1 observed vs expected values ............................... 50
5.4 Task 2 observed vs expected values ............................... 50
5.5 Task 3 observed vs expected values ............................... 51
5.6 Tasks 1-3, chi-square and p-values for high intensities ....... 51
5.7 Task 4 observed vs expected values ............................... 52
5.8 Task 5 observed vs expected values ............................... 52
5.9 Tasks 4 & 5, chi-square and p-values for high intensities .... 53
5.10 Probabilities of n correct pairs .................................. 53
5.11 Task 6 observed vs expected values ............................... 54
5.12 Previous experience and difficulty ................................. 55
5.13 Likert scale distribution ........................................... 55
5.14 All correct and previous experience distribution ............... 56
5.15 Tasks 1 and 2, observed vs expected values .................... 63
5.16 Contingency table for tasks 1 and 2 observations ............ 63
5.17 Previous experience and difficulty votes ......................... 64
5.18 Contingency table for tasks 1 and 6 ............................. 64
Preface

The work on this thesis started in the autumn of 2019 and lasted through the spring of 2020. The university was closed for the last three months of this period, which lead to a sudden halt of the implementation and experiments due to the necessary equipment being unavailable. Luckily, the most crucial work had already been done at that point, making it possible to complete the thesis with some adjustments.

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor: Kyrre Glette, for his guidance, support, and encouragement.

I owe a huge thank you to doctoral research fellow Çağrı Erdem, for helping me with the experiments involving sound.

In addition, a thank you to postdoctoral fellow Ulysse Côté Allard, who provided me with code to make essential hardware equipment work.

A special thanks go to my fellow master students at room 4104, for making each day at the university a joy. The long days, fun conversations, and lovely lunch breaks will not be forgotten.

Last but not least, gratitude goes to all my friends and family who supported me throughout this thesis.
Chapter 1

Introduction

In pursuit of attracting new audiences, augmented reality (AR) has been used by the entertainment industry in various fields such as concerts, theatres, museums, and amusement parks. The use of AR in entertainment is still in an experimental phase, but with time the technology is getting more sophisticated and opens up new possibilities. There are many opportunities to explore, but designing and planning additional AR effects for a whole show or artistic performance is time-consuming and costly.

Many people are walking around with mobile devices in their pockets capable of rendering 3D graphics, making it possible to experience AR without any specialized hardware like headsets. However, creating an artistic performance suited for AR is where the hurdle lies because it requires more time, choreography, and rehearsal. This thesis attempts to solve this problem and investigates a method for automatically generating AR effects, making it possible to add dynamic effects to almost any live performance, using a gesture control armband that reads muscle activity.

The visual difference in muscle activations can be very subtle, some even not visible. In a performance like Erdem and Jensenius 2020, where the sound is generated from muscle activity, it can be difficult to understand the mapping between muscle activity and the sonification. Having visual effects that improve the perception of muscle activity could help understand the mapping and strengthen the interactivity in a performance. This thesis also investigates this and presents a visualization designed to enhance our perception of muscle intensities.

1.1 Thesis goals

The goals of this thesis are to:

- Develop an AR system to improve a live performance.
- Investigate if AR can improve the viewer’s perception of muscle intensities.

1Video of performance: https://youtu.be/_-dzA5pl9k
The corresponding research questions are formulated below:

- Can AR improve a live performance?
- Can AR improve the perception of muscle intensities?

1.2 Contributions

The main contribution is showing that it is possible to create a system that can be applied to most artistic performances, along with a visualization method that makes it easier for an audience to perceive the muscle activity.

1.3 Thesis structure

This thesis consists of six parts, Introduction, Background, Software and tools, Implementation, Experiments and results, and Discussion.
Chapter 2

Background

This chapter will cover the required background knowledge for the implementation of this system and give a general overview of the field of virtual and augmented reality and its use cases, and related work.

2.1 Virtual reality

Virtual reality (VR) uses a headset with a built-in screen that displays a three-dimensional virtual environment in which you can interact and look around freely. The idea of a virtual reality through a head-mounted display (HMD) started already in 1965 and was presented by Ivan Sutherland [2], and he realized the first VR HMD in 1968 [3]. The technology has dramatically improved since then, and HMDs have reached the consumer market, making the technology available for more people.

In a virtual environment you have no reference point to the real world, so you need a way to determine where in the virtual world you are. A standard method for this is to track the user’s movement using built-in trackers in the HMD using lighthouses (fixed base station). When using a HMD as a tracking system, the desired outputs are the pose (position and orientation) of the individual trackers, and the inputs to the system are inertial measurements and light data. There are two types of light data emitted by the lighthouse, synchronized flashes, and infrared sweeping-planes, seen in figure 2.1. Each cycle starts with the synchronized flashes followed by horizontal and vertical infrared sweeping planes, and the time difference from the flash to when the sweeping plane hits the tracker is used to estimate the angle from the lighthouse to the tracker. By using two of these lighthouses, the system can determine the tracker’s absolute pose and then know where in the virtual world the user is.

Today the main usage field for VR is in the entertainment industry. However, there are many more use cases for VR. It can be used for remote collaboration, surgery support, or in the construction industry to visualize projects, among other applications. Visualizing data graphically in 3D could help us understand the data in a more intuitive way. Some of the
biggest challenges when it comes to data visualization in VR are: 1) how to visualize the information in an understandable and meaningful way, 2) how does the human perception error affect the virtual experience.

Reviews of independent studies show that users misjudge distances in virtual environments on average to be 73% of the intended distance [4], while in real environments, the users on average judge correctly. Kelly et al. [5] investigated if this is still the case with modern HMDs using data from previous studies [6][8]. These results show that distance under-perception in VR is still an issue, but the problems seem to vanish as technology improves. In a study where human perception error is low in both desktop and immersive-based conditions, the immersive experiences required less effort and navigation to find the correct information, and also give a higher subjective perception of accuracy [9]. However, when it comes to task completion time, the 2D desktop was significantly faster than 3D immersive conditions in all cases.

2.2 Augmented reality

In contrast to VR, AR acts as a digital addition to the user’s world rather than replacing it. AR is about changing how we perceive the physical world around us by adding layers of digital information like computer graphics and sound. This can be achieved by using cameras on mobile devices, or HMDs that use semitransparent mirrors, and various other methods. The idea of AR was first conceptualized in 1992 when Thomas Caudell and David Mizell, two scientists at Boeing, was tasked with helping assembly line workers with long wire bundles. They published the article Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes [10], where they wondered if they could give their workers a see-through display with overlaying computer graphics telling them where the wires should go. This early attempt at AR did not catch on, and the field saw little development for over a decade, but this changed with the arrival of smartphones. The processing capabilities, camera, and portability of the smartphone have made it a popular platform for AR development, and applications with AR features like Instagram,
2.2.1 Augmenting the world around you

Exactly how different applications achieve AR can differ a lot. One of the most simplistic forms of AR is the marker-based type. Markers on a flat surface that is identified by a camera are used to anchor the digital world to the real world. Any distinctive image recognized by the camera is called a marker. This means that any picture can be a marker as long as it is unique enough. Marker-based AR is mostly used when we need to know precisely what the user is pointing at with their camera. An example of this could be looking at a poster with a marker, and then 3D models related to that poster would pop up in AR.

In other cases, a marker might not be the best choice. If we want to provide walking directions or road names, local information has to be provided. This is known as location-based AR. Users can use this type of AR to get information when they are walking in unfamiliar streets and even display directions on top of the physical roads. Using both the GPS and the compass sensor found in most mobile devices, virtual objects can be placed with high accuracy in large areas such as streets, parks, and airports.

Some online shopping applications use AR to let customers place virtual furniture in their homes, as a way to preview the item, like in figure 2.2. In these types of applications, the user can test out different objects and locations in their room. For the user to be able to place the virtual object in any arbitrary room, the floor area has to be detected. The mobile device also has to calculate its pose relative to the room when moving around. A well-known method for this is simultaneous localization and mapping (SLAM) [11, 12], or more specifically for camera-based applications: visual simultaneous localization and mapping (vSLAM) [13]. When the mobile device is moving around, the algorithm detects visually distinctive features called feature points that are used to calculate changes in the pose over time. These features then have to be robust, meaning they must be unaffected by changes in the camera perspective, rotation, scaling, and lighting.

Further, these feature points can be used to look for planes in the real world, by looking for groups of feature points that lie on the same vertical or horizontal surfaces, such as feature points on the floor or a wall. Detected planes then act as a bridge between the virtual world and the real world, as they give virtual objects something to anchor onto. The virtual camera that renders 3D objects is also aligned with the pose of the mobile device, making sure the virtual objects are rendered from the correct perspective. Virtual content then looks a part of the real world, as the rendered virtual objects are overlayed on the images from the mobile

1Checked January 2020: https://play.google.com/store/apps/top
2.2.2 Applications of augmented reality

Educational use

The human perception is three-dimensional, and we think and store information in three dimensions in our brain [14]. When we look at a drawing on a flat piece of paper that is meant to look like it is three-dimensional, it takes some time for our brain to process how it would look in 3D. Some studies [15–17] use AR for students to directly see and memorize 3D anatomy structures, helping them be able to learn complex anatomy structures faster than with traditional methods. Because of the spatial visualization and interaction possible with AR, it can provide a better learning environment in a classroom situation. Some students who struggle in a traditional text-based learning environment can benefit from this type of learning environment [18]. Other studies also show that the perceived enjoyment of the AR experience is a key determinant in the positive effects of such applications, rather than the perceived usefulness [19].

Surgical use

Using AR as a surgical tool is found to be a positive experience by some [20][21]. Displaying 3D holograms models of a heart before surgery can help to understand the complex morphology, and it can also be used as a method for surgical navigation [22]. The focus must be on the patient during surgery, so the additional graphics must not be a distraction. AR during a procedure can also obstruct the view, and advanced navigational displays may increase precision, but can make it less likely to identify significant unexpected findings clearly within sight [23]. Therefore, it is important to acknowledge that AR does not enhance all surgical situations, but can be a powerful tool when used correctly.
Human-robot interaction

AR has seen use in the factory and manufacturing industry \cite{24, 25}, and in an industrial assembly line where cooperation with industrial robots is important, extra visual information can be useful. In a case study with a high payload robot \cite{26}, the robot performed automated tasks while the operator did more delicate tasks like managing the cables, avoiding any collision with the robot. The goal is for the operators to receive information for each production step, visual and audio warnings, robot motion, and workspace visualization. These experiments have validated the applicability of systems using handheld devices, and also shown to enhance the operator’s safety awareness.

Enhancing creative experiences

A widespread use for AR is as an entertainment tool, despite not directly being a practical field of application, it can be used as an alternative way of expressing creative skills. An example of this is that in recent years, AR has been used to bring flat children’s drawings to life. The idea behind these types of applications is that since children spend an increasing amount of time absorbing passive content through television and digital devices, an interactive coloring book could help children become more engaged in real-world creative activities. Given simple drawings as inputs, applications can generate interactable 3D models overlayed on the real world, and show increasing motivation for children to draw more \cite{27–29}.

2.3 Related Work

2.3.1 Augmented reality in dance performances

Younger people have become less interested in traditional dance productions \cite{30}, and choreographers, dancers, and producers are trying to find ways to attract a younger audience. Additional visuals and sounds in interactive performances have been used in attempts to engage younger audiences. Dance performances with pre-rendered videos and images showing in the background are not uncommon these days. The choreography is synchronized with the video, giving the illusion of interaction with the dancers. Others have tried to explore this type of illusion using the performer’s body as a canvas to display pre-rendered videos and images \cite{31}. However, these examples are not truly interactive performances.

AR combined with a tracking-based system can be used in multimedia performances, offering interactive visualization for live performances. Designing a live interactive performance can be difficult as it has to engage both the performer and the audience. Some suggest that user-centered design with the addition of aesthetic perspectives should encourage improvisation and intrigue the user \cite{32}. A dance performance is very much open for improvisation, but the challenge is having digital content that also
is open for improvisation. Clay et al. [33] solve this by having a large screen behind the dance performer and capture the performer’s movement using motion capture. By tracking the performer, they were able to display virtual shadows, have the performer manipulate a virtual cube, and dance with avatars mimicking the movement. They also tried to recognize emotion based on the dancer’s movement, but this proved to be difficult for the audience to perceive. This led them to the conclusion that all augmented visuals and audio must be open for interpretation for the audience. Simple direct interaction like manipulating a cube worked much better when presented to an audience, based on the feedback.

Capturing the performer’s position and movement is essential in these types of showings, but often require covering people in markers to be able to capture them. A markerless tool for capturing motion and gestures is Microsoft’s Kinect, and it can capture positional data and skeletal bodies in real-time, using its 3D depth sensor camera. Widely applicable techniques for the gaming and film industry could be used [34], but the camera sensor area is roughly 6m². The somewhat small capture region limits the production, making it unfeasible for traditional performance stages, which are greater in size. Another type of markerless motion capture is called Organic Motion [2], it uses multiple cameras from all angles like ordinary motion capture, but require an easily separable background. This creates an encapsulated area, like a 360° green screen, making it unsuited for a stage.

For markless motion capture better suited for a stage, Brockhoeft et al. [35] have created a setup using infrared light emitters and a camera that can detect light in the infrared spectrum. They show a prototype system that successfully integrated into a public dance performance, using the infrared light to segment their form from the background. Dynamically generated visual effects are displayed on a screen behind the dancers in real-time, which follows the position of the dancers, enhancing the expressive power of the performing artists. What could be a problem in this situation is that the audience might not realize that the visual effects are dynamically created. From an audience perspective, it could have been a pre-rendered video in the background, with a rehearsed choreography.

Although some try markerless options for motion capture, the marker-based systems have evolved into very mobile, specialized, and simple to use ones [36]. It is highly robust and can be used to capture difficult-to-model physical movement, that would otherwise require skilled animators and be time-consuming, making it a great tool for VR and AR. However, Markers on the body might restrict movement, which must be taken into consideration.

2.3.2 Performance interaction using mobile devices

AR has also been used to create audiovisual performances that allow the audience to interact with the work presented by a performer through their mobile devices. Audience members can perceive virtual objects and effects on their mobile devices, by pointing their device towards the stage, having large images behind the performers that the AR application recognizes [37]. Spectators can manipulate the music by interacting with AR elements in the application, and the idea is to increase the audience reward by letting the audience participate. The performance design also allowed for a non-participating audience by using a large projection screen to show the stage augmentations to the whole audience. The performance was an acoustic rock band, and they show positive feedback through questionnaires. A downside to this performance’s design is that the performers have to stand still, not to obstruct the view of the image targets in the back.

Interactive AR changes the passive audience to active participants, and in a live performance using AR on mobile devices [38], it was found that holding up a mobile device during the performance was exhausting and fatiguing. The artist used a gesture-controlled virtual instrument and four large visual markers on the sides of the stage. These 1x1 meter AR markers define a visual instance (recorded video clip) of the performer that appears over the marker. The artist recorded movements over a certain time interval during the performance, which creates the visual instances, creating multiple layers of video and audio content. The audience reported that they only watched the AR content for brief periods relative to the total length of the performance, as holding the mobile device was tiresome. A solution to this could be to provide cheap cardboard glasses for the smartphones and create a see-through mode for the application, but this would also disable interaction with the mobile device.

Other ways mobile devices have been used is by voting for playing modes [39] communicated to performers through live graphic scores [40]. In this setup, the performers and the audience faced each other in a traditional performance setting. Behind the performers was a large projection screen displaying graphical symbols representing the playing modes. Results from their surveys showed that participating performers and audiences valued the process of musical creation sharing. The audience took creative decisions and decoded their effects, and the performers followed the generated scores, which was challenging for both groups. Positive responses came mainly from the audience participants who felt challenged, as it engaged them more and made them feel close to the performers. Some misunderstood the system or wished for more control, leading to some frustration.

Fazekas et al. [41] created a similar performer and audience interaction system for live improvised music using mobile devices. Instead of defined playing modes, the participants can vote for emotion coordinates in a
2D arousal-valence space (calm-excited, negative-positive). The audience’s emotional directions were visualized as colored spherical blobs on a projective screen behind the musicians, and the size of the blob represents the number of users indicating an emotion cue in the same 2D area. They found that their system parameters are sensitive to the number of participants, but by recording data during concerts, a parameter database could be created. This allows for replaying of the data during rehearsals, and the best parameters could be determined for different audience sizes by analyzing the data.

Recording live performance data appears to be useful, as it can be used to improve a system further. Recorded user data could also be valuable for other music and human-computer interaction studies.

Kayali et al. [42] discuss a setup for technology-mediated audience participation (TMAP) in live performances. Their system allows for internet connection free communication using high-frequency sound IDs to trigger music samples and colors on the audience’s smartphones. Audience members can choose a location in the performance space, and together with parameters selected by the performer, creates a soundscape. From the performer’s perspective, the learnings are that a wide variety of functionalities are not necessary, as constraints encourage creativity. Low latency, reliability, and sound quality seem to be what matters the most to the artist. The artist also wanted full control over the musical result, and balancing this with freedom for audience interaction was one of the core challenges.

### 2.3.3 Real-time human arm tracking

Visualizing virtual objects around human arms in a realistic way is challenging as you need to track the exact pose in real-time. Gunes et al. [43] present a markerless AR application for virtual accessory around human arms. They utilize a Kinect sensor to capture depth, standard RGB video, and it also provides skeleton joint poses. The width of the arm is calculated using the RGB and depth camera, and the Kinect finds the correct orientation and location. Further, OpenGL (a multi-layer rendering framework) is utilized to real-time render the 3D accessory and video together. To give it a more realistic look, fragment shading is used to occlude the part of the virtual object that is wrapping around the arm. As presented in the results, the system works in the 0.5m to 2m area away from the sensor. Unfortunately, observations also showed that the depth data became extremely noisy when the arm was too close or in front of the body, which restricts the wearer’s arm movement.

Arm movement can easily be detected by cameras and be precisely measured using a motion capture system, but what is not visible is the muscle activations. The muscle contractions in our body generate motions, and some studies suggest visualization methods for this muscle activity. Murai et al. [44] have created a system that visualizes muscle tension
information in real-time using motion capture and electromyography (EMG). A rendered model is displayed on top of images from a standard camera with precise estimation by combining a physiological muscle model with inverse dynamics. A large amount of computational power is required for this system, resulting in a system that runs at nearly 15 fps.

Kishishita et al. [45] propose a low-cost method for visualizing muscle effort in human arms. A motion capture system (Optitrack) and software for biomechanical modeling (OpenSim) was used to model muscle activities using linear interpolation. The muscle activities were then visualized on a screen with AR methods and markers. One marker represented the base position, and another marker was attached to the user’s hand to capture its position. The distance and orientation between the two markers were used to register the arm posture; one arm posture is one position in the 3D space. The visualization consisted of a single cube with color changing depending on the muscle effort, at the position of the marker following the arm, representing the subjective effort of the motion. What is not discussed in this study is if only having changing colors is a good enough visual representation of total arm effort.

2.4 Statistical method

This section covers the statistical methods used in the experiments chapter.

2.4.1 Null hypothesis and alternative hypothesis

The null hypothesis, denoted $H_0$, is considered the default in a scientific experiment and is stating that there is no difference in two measures. The alternative hypothesis, usually denoted $H_A$, is stating the opposite, that there is a difference. Using statistical methods, the null hypothesis can be rejected, and thus the alternative hypothesis is accepted. If the null hypothesis is not rejected, it does not mean it is true, but it only means there is not enough evidence for the alternative hypothesis.

2.4.2 P-values

The p-value is a probability value that essentially is a measure of how extreme our sample is under the null hypothesis. For example, if the null hypothesis states that there is no difference between two populations, but our sample has quite a big variation, then the p-value tells us how extreme our sample is if we presume it came from a population of equal proportions. If our p-value is 0.02, then the chance of getting the sample we got is 0.02% if we assume the null hypothesis is true. A p-value of less than 0.05 is considered statistically significant and means the null hypothesis should be rejected.
2.4.3 Chi-square test

The chi-square goodness of fit test is a method to measure the statistical significance of a hypothesis where categorical data is compared and each observation is independent. This test is usually performed on a contingency table, which shows the frequency distribution of categorical variables. A test identical to the chi-square test is the z-test, except that the standard normal deviate is calculated instead.

2.4.4 Mann-Whitney U test

The Mann-Whitney U test is a test where two groups are compared without the assumption of a normal distribution, to test if the distribution of two populations have the same shape.

2.4.5 Likert scale

The Likert scale is commonly used in surveys, and is a rating system where the responders typically select from five points to specify their level of agreement. So, for example: 1 = strongly disagree and 5 = strongly agree.
Chapter 3

Software and tools

This chapter will cover a brief introduction to the software, tools, and equipment used in the implementation of this project, such as the game engine used, the motion capture system, and the AR platform used.

3.1 Myo armband

Myo is a gesture control armband worn on a user’s forearm. It has inertial measurement units (IMU) and eight surface electromyography (EMG) sensors. Using these sensors, the Myo can detect the user’s arm movements and hand gestures, and connect to other devices using Bluetooth. There are five pre-set gestures that the armband can recognize: wave left, wave right, double-tap, fist, and fingers spread. The raw EMG data can also be accessed as unitless values for each sensor, representing intensity.

The developer of Myo, Thalmic Labs, ended the sales in 2018. Therefore, several of their websites no longer exist.

3.2 OptiTrack & Motive

OptiTrack is a motion capture provider, and the product line includes software and high-speed cameras. It sees use in robotics, virtual reality, games, and film such as Disney’s The Lion King. Motive is an optical motion capture software that collects and processes motion capture data from OptiTrack cameras. The OptiTrack system used in this thesis is 12 synchronized 100Hz cameras (Figure 3.1) that are installed on the walls and ceilings. Each camera tracks 2D positions, and these 2D positions are measured against each other to calculate 3D positions using triangulation. This system is capable of capturing human-body and rigid body movements with high accuracy, and also live stream this data to other software in real-time.

1https://medium.com/@srlake/ending-sales-of-myo-preparing-for-the-future-281af9bbca2
2https://optitrack.com/about/press/20190910.html
3.3 Unity

Unity is a cross-platform game engine, mostly known for game development. The free version of Unity is available to everyone, and this has led to a massive online community for assistance and tutorials. Unity has extensive documentation and can deploy to a wide range of platforms.

3.3.1 Scripting

An essential part of making applications in Unity is scripting, and it uses C# as its only programming language. In Unity, everything runs as single discrete frames while a game scene is running, and the execution of these frames happens as fast as Unity can manage. All scripts derive from the base class `MonoBehaviour`, and an essential part of scripting in Unity is event functions that are inherited from `MonoBehaviour`. When an event triggers during gameplay, Unity passes control to a script and its functions until it has finished executing. The two most commonly used event functions are the `Start` and `Update` function. `Start` is mostly used for initialization and `Update` is the place for code that is to be executed every frame. Scripts have a predetermined order of execution for event functions, and the order of execution in figure 3.2 shows that `Start` is only called once and that `Update` is called each frame as a part of the game logic.

3.3.2 Objects

Scripts can be attached to objects, and the fundamental objects in Unity are called `GameObjects`. The scripts attached tells the object how to behave, as figure 3.3 shows, multiple components are usually attached to a `GameObject`. The component `Transform` must always be attached, as this

---

3Recreation of the image found on https://docs.unity3d.com/Manual/ExecutionOrder.html, 29.01.2020.
Figure 3.2: Order of Execution in Unity
represents the position, orientation, and scaling of the object in a scene (A scene in Unity can be looked at as a level in a video game). Each object can have a parent, and the hierarchy of the objects is important, as the Transform can be relative to world space or relative to the parent object.

3.3.3 OptiTrack plugin

The OptiTrack Unity3D Plugin is used to capture streamed real-time rigid body data from Motive. Attaching the OptiTrack rigid body script to an object in Unity makes the object mimic the movements of the real-world object tracked by Motive. The positional data received should be treated as local positions in Unity and not global positions, as the origin and orientation in Motive are unlikely to match the global origin and orientation of Unity. Using the data received as local positions will reflect the object’s relative position to its parent object. Organizing it this way makes it possible to adjust the positional data from Motive by transforming the parent object in Unity.

Figure 3.3: A green sphere in Unity with the script seen in figure 3.4 attached.
using UnityEngine;

public class MoveBall : MonoBehaviour {
    // Use this for initialization
    void Start () {
        // Set the start position (x,y,z)
        transform.position = new Vector3(0, 0, 0);
    }

    // Update is called once per frame
    void Update () {
        // Move the ball one unit on the x-axis every second
        // Time.deltaTime is the time it took to complete the last frame
        transform.position += new Vector3(1*Time.deltaTime, 0, 0);
    }
}

Figure 3.4: An example of a simple script in Unity. A common way to change the position of an object in Unity is to use transform.position.

3.4 ARCore

ARCore is Google’s software development kit (SDK) that enables developers to work with AR on Android, iOS, Unreal, and Unity. The core features of ARCore is tracking the mobile device as it moves around, and the creation of its understanding of the real world. This is accomplished using some essential concepts:

- **Motion tracking** - Position of the mobile device relative to the world.
- **Environmental understanding** - Detection of surfaces like floors, walls, and tables. Including their size and location.
- **Light estimation** - Light the virtual objects according to the lighting condition in the current environment.
- **Anchors** - Ensures that virtual objects appear to stay in the same position over time.

With these concepts, developers can enable their applications to overlay features onto real-world images. To start working with ARCore, developers must download the SDK for supported platforms\(^4\).

ARCore is not the first AR platform from Google. When ARCore was released, they ended the support for their other AR platform, Project Tango\(^5\). Tango did not catch on because it required additional sensors and new hardware for the mobile devices. ARCore supports what smartphones

---
\(^4\)https://developers.google.com/ar/develop
\(^5\)https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality
already have, and can not be expected to be as accurate as Tango, but working with your current phone is often an advantage. A disadvantage of ARCore was the lack of occlusion detection, but now the ARcore Depth API is in the works, which uses the existing monocular color sensor in mobile devices to create dense depth maps.

When working with ARCore, Unity, and Android, there is an additional application called Instant Preview that can make the development process more manageable. Instant Preview makes it possible to skip the build process in Unity to instantly test the ARCore application on the Android device by connecting it with a USB cable.

3.5 Google Cloud Platform

Google Cloud Platform offers a wide range of cloud computing services. In this thesis, the platform is used to enable cloud anchors using the ARCore Cloud Anchor API. Using cloud anchors, multiple people can view the same virtual object placed in a physical location in an AR scene. After an anchor is locally created, ARCore can upload that anchor as a 3D feature map to the ARCore Cloud Anchor service, which creates a cloud anchor with a unique ID. With this unique ID, other users can recreate the same anchor and share the same AR experience, as seen in figure 3.5.

Figure 3.5: Cloud anchor connecting a virtual object to the real world

Chapter 4

Implementation

This chapter will cover the implementation process for this system. The system consists of several parts that were worked on individually before put together into a complete system.

4.1 System development

4.1.1 Prototype

This section covers how Unity was utilized to achieve AR and the prototype for a shared AR experience. The prototype was created and tested using two Android phones, a Myo armband, and a computer running Unity.

Augmented reality in Unity

Unity integrates a platform called Vuforia Engine in version 2017.2 and later. This is an AR platform used only in the prototype. The main feature of Vuforia is image tracking, and it is a quick way to get started with AR in Unity. Vuforia combines image tracking with object tracking; this enables the use of non-flat images. By uploading a target image and selecting a target shape in the Vuforia target manager, one can create a target GameObject in Unity. A cylinder GameObject with a diameter of 8.8cm and a height of 6.0cm and a white image with black lines was created using Vuforia, as seen in figure 4.1a. The idea was that this cylinder could be placed on a person’s wrist like an armband to track their arm. A real-world copy of this cylinder also had to be made, seen in figure 4.1b.

All active GameObjects in Unity that have the object in figure 4.1a as a parent will be shown through AR if Vuforia can track the real-world object in figure 4.1b.
Communicating with the Myo

The initial idea was to have each device running the application connect directly to the Myo via Bluetooth. It did not take long for this to become a problem since the Myo was only able to connect with one mobile device at the time, and this sparked the idea of socket programming.

By using sockets to send UDP(User Datagram Protocol) packets over Wi-Fi, messages can be sent wirelessly from any Wi-Fi enabled device. Sending messages from the broadcast address[1] will send out to anything that is listening on the same Wi-Fi and port. Since this is a real-time system, it is preferred to drop packets rather than waiting for delayed ones, hence the choice for UDP.

This process had some obstacles, sending broadcast messages on a private Wi-Fi should not be a problem, but the university Wi-Fi has restrictions. The imperfect solution to this was to set up a mobile Wi-Fi zone using 4G from one of the mobile devices. The mobile Wi-Fi has no problem sending broadcast messages, but having all the devices connect to it can be a bit time consuming, as most devices already are connected to the university Wi-Fi.

In this setup, a computer is working as a hub and is sending out the

---

1IP address 255.255.255.255
Myo status messages as broadcast messages. For the prototype, the messages are strings that describe the hand gesture from the Myo; this could be Rest, Fist, WaveIn, WaveOut, FingersSpread, DoubleTap, or Unknown. UDP packets consist of bytes, and therefore the strings that describe the hand gesture has to be converted to bytes before sending. On the other side, something has to deconvert and read these messages. In Unity, two scenes were created. One scene handles the sending of broadcast messages and is ran on the hub. The other scene reads the messages, and this is the scene that runs on mobile devices.

For the testing of this system, some animations were added to the Unity project. These animations were then associated with a specific hand gesture from the Myo, meaning a hand gesture would trigger a particular animation for all the user running the application. The animations are instantiated with the cylinder target object as a parent, which means that the position of the animations as they appear is relative to the cylinder target. Figure 4.2 shows the animations appearing and is the first proof of concept for this shared AR experience. Creating a prototype addressed both the foreseen and unforeseen technical challenges of the full system, helping with further development and decisions.

Figure 4.2: Animations appearing with an offset in the Y-direction relative to the armband. Left shows flames appearing when spreading the fingers. Right shows an explosion when clenching a fist. The Myo armband is placed further down on the forearm.

4.1.2 Selecting a tracking method

Using the Vuforia cylinder tracker would not last as a permanent solution, as the software struggles to track the cylinder at distances above roughly 1 meter. Other ideas had to be tested to conclude on a final method of tracking a person.
OpenPose

OpenPose is a real-time system that detects human body, foot, hand, and facial keypoints from images [47–50], and OpenPose Unity Plugin is a wrapper for this system. The idea was to use these keypoints to locate the arm of the person wearing the Myo. Testing this in Unity did not turn out to be a success, as the plugin struggled to find the human pose both close and far away. While their demo video for the plugin shows somewhat stable tracking, the framerate rarely goes over 6 FPS when it is tracking a person. Running this on mobile devices would then be expected to have even lower framerates, so other methods for tracking had to be explored.

Pre-trained machine learning models

Machine learning models can be used to detect objects and different classes of objects. To see if this could be used as a tracking method was tested using TensorFlow inside Unity. Getting TensorFlow to work in Unity is not a straightforward process. Luckily, an active person in the Unity community solved this and shared their code for others to use. This example use the pre-trained object detection model ssd_mobilenet_v1_coco_2017_11_17, that is trained on the MS COCO dataset [51]. It has 90 types of objects that the model can detect, and Person is one of these classes, so it is possible to use this model for a proof of concept system.

The model detecting a person can be seen in figure 4.3 and since it detects a square area with a person inside, it is not precise enough to say exactly where the person is in the image. While other machine learning models better suited for this AR system probably could be tested out, another problem would still occur, placing virtual objects with a fixed size would not be accurate. Using the detected person’s size in the 2D image would not be an accurate description of measurements in the real world, and there is also no way to obtain the orientation of the person. This case would also apply to the OpenPose system. The virtual object’s size in the image could be set relative to the tracked person’s size in the image, but can not be set relative to real-world dimensions. To create an immersive experience, a way to track real-world measurements is needed, and thus the idea to use image-based object detection was deprecated.

Motion capture

The precision and robustness of motion capture is a strong point when it comes to tracking. However, the drawback is that it limits the AR system to a specific room or area with motion capture cameras installed. After some thought, having a system that can almost perfectly capture real-world movements outweighs the location-based limitations, and thus became the

2https://github.com/CMU-Perceptual-Computing-Lab/openpose_unity_plugin
3https://www.tensorflow.org/about
4https://github.com/MatthewHallberg/TensorFlowUnity-ObjectDetection
chosen tracking method. The challenge then becomes knowing where the mobile devices are located in the room relative to the motion-tracked object. One solution could be to put reflective markers on the mobile devices as well, tracking both the person and the phones. This way, the necessary real-world distances and angles are known. The scalability of this would then be in question, as each mobile device has to be tracked, as well as be in sight of the motion capture cameras, leading to a crowded floor if there are many users. Mobile devices should not have to be inside the motion capture area, but even so, still have a known position in the room.

4.1.3 ARCore & motion capture

We want a system where each user can point their mobile device at a motion-tracked object and add overlying graphics to that object, as shown in figure 4.4. For this to be possible, the motion capture coordinate frame and the virtual coordinate frame has to match. As a result of trying to solve this problem, the ARCore AR platform became an essential part of this system.
Cloud anchors

The base for this system is the CloudAnchors sample scene that comes with the ARCore SDK for Unity. In this sample, seen in figure 4.5, ARCore tracks the position of the mobile device, using the camera to identify feature points in the physical space. It also detects planes and includes a room sharing server-client, and cloud anchor controllers.

One could expect that cloud anchors would provide the full multi-user experience, meaning that once a cloud anchor is hosted, it would be easy to update the anchor and have them continuously be resolved on all the mobile devices. While experimenting with cloud anchors, this did not seem to be the case. ARCore works perfectly well for instantiating static models like a figure or a picture. However, to create a real-time experience in AR using ARCore can be challenging, as changes to the cloud anchor are not instant. Hence, the processing needs to be done on the mobile device for it to work in real-time.

The idea then became to use the anchor object as a pseudo world coordinate frame, seen in figure 4.6. Since the anchor can be placed with high precision, it is possible to place it at the point in the real world that is the origin of the motion capture system. Then the placement of
the anchor object and the origin in Motive is roughly the same, at most a few centimeters in variation. If the placement and the origin were to be precisely the same, and ARCore’s understanding of the world is theoretically perfect, then the behavior in figure 4.4 can be achieved. By creating an object in Unity that is a child of the anchor object and setting its position to the position of the tracked real-world object in Motive, the virtual object would mimic the tracked object’s movement. As long as the anchor object is 1 unit in size, and the rotation of the anchor object matches the axes of the Motive coordinate frame, then the desired behavior could be achieved.

Figure 4.5: CloudAnchors sample scene. 1: Moving the mobile device around makes ARCore able to detect a plane. 2: Selecting ‘Host’ and pressing on the detected plane places the anchor exactly where you pressed. 3: After the anchor is hosted, other users can resolve the anchor by entering the room code and IP address, making them see the same as the host in 2.

Sending data from Motive

For this to be possible, the positional data from Motive is needed. Sending the data to all the mobile devices is done the same way as sending the Myo data, though broadcast messages. The Unity scene running on the hub contains a cube and a sphere. Here the sphere is an OptiTrack client object, meaning its position equals the positional data received from Motive. Since it is local positions, the sphere has the cube as a parent object. This makes

\footnote{Default value in Unity is 1 unit = 1 meter, which is used in this project}
the sphere move relative to the cube. If the motion-tracked object is moved 1 meter in a direction from the center, then the sphere in Unity moves 1 meter away from the center of the cube. The position of the sphere is sent as broadcast messages and received by the scene running on the mobile devices. In this scene, there is an identical sphere and cube setup. Here the cube is the cloud anchor object, and the sphere receives its position from the broadcast messages. Placing the cloud anchor at the origin of the motion capture system should then make the motion-tracked object and the sphere move correspondingly.

Testing ARCore & Motion capture

The origin of Motive is decided during the calibration of the system, as is the orientation. By placing the ground plane calibration triangle shown in figure 4.7 on the floor, the origin and orientation are chosen. A red dot on the floor marks the origin because it is crucial to know where the spot is for later use. Further, a motion-tracked object is also needed. This is the same armband as in figure 4.1b, but with some reflective markers added to it with velcro.

After ARCore has detected the ground plane like in figure 4.5 and host mode is entered, the anchor object then has to be placed at the red dot. This setup process for matching the motion capture coordinate frame and Unity object’s coordinate frame is shown in figure 4.8. When the anchor object is placed, the orientation is most likely going to be wrong, so the green cube has to be rotated. Rotating the object is manually done by swiping on the device screen, and you then have to rotate it so that the blue sphere is lined
up with the motion-tracked object on the floor. Calibrating the rotation is entirely done by eyeballing it, looking at it from different angles is needed to make sure the sphere and object are completely lined up. So that each user does not have to rotate the object, the anchor is locally instantiated and then uploaded as a cloud anchor after its rotated, giving the anchor the correct rotation for the other users when they resolve it.

After the manual calibration, the motion-tracked object can be moved around, and the sphere appears over it. An example of this is in figure 4.9, where the motion-tracked object is kicked around, and the sphere follows over it. This is happening in real-time as the positional data is received from the hub.
Since the manual calibration now is seen to be working, the green cube is set as invisible, as it serves no visual purpose. To further test the system and discover its potential, a trail renderer was added. The trail was set to permanent, giving the effect of being able to draw real-time in 3D space, seen in figure 4.10. Even though this is not the intended use case of the system, realizing the capabilities of the system was important to develop the system further.

Figure 4.10: Using the system to draw in 3D space. Videolink: [https://youtu.be/ONrchCUxEwQ](https://youtu.be/ONrchCUxEwQ)

---

6https://docs.unity3d.com/Manual/class-TrailRenderer.html
4.1.4 System overview

With the combination of ARCore and motion capture in place, the core system is complete. Figure 4.11 gives a visual overview of the communication between the components and how the base system works at a simplified level. The next step is to design a visualization system on top of this using the Myo data.
Figure 4.11: The core system.
4.2 Visualization and user experience

4.2.1 Particle systems

There are several ways to create animations and effects in Unity. Meshes (3D) and sprites (2D) are often used to create animations, but these are better at representing solid objects, such as a playable character moving around. Another method for animations is particle systems, which are useful for creating dynamic objects that are difficult to create from meshes and sprites, such as liquids, smoke, or fire. The Built-in Particle System in Unity simulates the particles on the CPU, making it possible to run on every available platform in Unity, and use the same physics system as everything else in the scene. It can also be used to create custom effects from scripts, giving full read and write access to the particle system during runtime. Unity’s Built-in Particle System hence became the particle system of choice. Another particle system called Visual Effect Graph exits, but this is a solution that runs on the GPU for large-scale visual effects, therefore automatically ruled out as it is not compatible with mobile devices.

The particle system is going to be a visual representation of the Myo data. Since the streamed data from the Myo represents intensity, the main idea is to have fewer particles and little particle movement at lower intensities, and more particles and motion at higher intensity values. The particle system was then set to emit particles where the particle amount and particle velocity uses the intensity as a factor. Because the Myo has eight different channels to extract raw EMG data from, the initial idea was to use eight different particle systems, and one to one map each channel to a particle system with different colors, seen in figure 4.12. Applying this visualization to the application, shown in figure 4.13, reveal some disadvantages. For example, it is difficult to tell which channel is the most active one of the red and the orange. Also, since the particles spread out in different directions, it makes the visuals appear a bit chaotic.

Intensity values from the Myo oscillate and is a bit noisy, which causes spikes in intensity and rapid changes in the colors of the visualization. To combat this, the outputted intensity not used directly, but the absolute value is taken and sent through a simple moving average filter (equation 4.1) with \( n = 5 \). This is applied to all channels and it stabilizes the values as the output \( \bar{X}_M \) is the average of the last five intensities for that channel, which smooth out momentary fluctuations in intensity. Since this is a real-time system, the moving average is always calculated from the latest timestep \( M \). The value chosen for \( n \) here is a trade-off between delay and stabilization. A large value of \( n \) flattens out noise, but require many timesteps for significant changes in the intensity to take effect, and swift hand-movements would then appear delayed. A small value of \( n \) is prone to noise but is better suited for real-time output. Testing of various values of \( n \) showed that \( n = 5 \) gave the best result for the visual effects, with a barely noticeable delay and minor unwanted effects due to noise.
Further, some substantial changes to the particle systems were needed to create cleaner visual effects. Still, the main idea with intensity as a factor for particle amount and velocity was preserved. The number of particle systems was reduced from eight to two, but two unique ones. The first particle system (A), seen in figure 4.14, is quite similar to the ones used in figure 4.13. The direction of the particle system was changed to go upwards, when wearing the motion-tracked armband, this is the direction from wrist to elbow. The color of the particles is also no longer predetermined, but chosen by the most active channel (see table 4.1). Particles last of a couple of seconds, so the channel with the highest intensity only sets the color of a particle the moment its emitted and does not change the color of already emitted particles.

\[
\bar{X}_M = \frac{1}{n} \sum_{i=0}^{n-1} X_{M-i}
\]  

(4.1)

The second particle system (B) is a radial particle effect, shown in figure 4.15. While it looks quite different, its behavior is similar to the other particle system. Color is chosen the same way, by the most active channel, and the number of particles increases with intensity, but it has a particle speed of zero. For this particle system, one circle is a single particle with a lifetime of four seconds, and the size of each particle grows linearly during its lifetime. The combination of the particle systems in figures 4.14 and 4.15 creates the visual effects seen in figure 4.16, which is the visual effects used for further testing. Parameters for both particle systems are shown in table 4.2 and figure 4.17.

<table>
<thead>
<tr>
<th>Channel index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Red</td>
<td>Magenta</td>
<td>Purple</td>
<td>Blue</td>
<td>Cyan</td>
<td>Green</td>
<td>Yellow</td>
<td>Orange</td>
</tr>
</tbody>
</table>

Table 4.1: Color and Myo channel correspondence.
Figure 4.14: Particle system A in the Unity editor, showing with increasing intensity from left to right. The gray cylinder represents the armband.

Figure 4.15: Particle system B in the Unity editor, with increasing intensity from left to right.

Figure 4.16: Particle systems A and B combined, augmented on the motion-tracked armband while wearing it.
Figure 4.17: The defined material used to render particles.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Particle system A</th>
<th>Particle system B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material</td>
<td>Figure 4.17a</td>
<td>Figure 4.17b</td>
</tr>
<tr>
<td>Start Lifetime</td>
<td>random [1, 2]s</td>
<td>4s</td>
</tr>
<tr>
<td>Start Speed</td>
<td>*</td>
<td>0</td>
</tr>
<tr>
<td>Start Size</td>
<td>random [0.2, 0.8]</td>
<td>5</td>
</tr>
<tr>
<td>Start Rotation</td>
<td>random [0, 360]</td>
<td>0</td>
</tr>
<tr>
<td>Particles Emitted/s</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Color</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Size over Lifetime</td>
<td>linear 1.0 to 0.5</td>
<td>linear 0.5 to 1.0</td>
</tr>
<tr>
<td>Particle Trail</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>Trail Ratio</td>
<td>1.0</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.2: Parameters of the particle systems. Unchanged default parameters are not listed. * indicates parameters controlled by values from the Myo armband.

An upgraded version of the motion capture armband was also made (figure 4.18), with more reflective markers that are screwed on to ensure a robust tracking. The markers’ exact placement is a result of trial and error and gave the most stable tracking from all angles tested.

Figure 4.18: The new armband, also covered in black fabric to make sure no parts except the marker are reflective.
4.2.2 Effortless connection

During testing, the room-sharing service included in the ARCore example was used as is, where the user has to input room code and IP address. Having many people input a code and IP could be a hassle, and since everyone has to be on the same Wi-Fi for the application to work, connecting through IP address becomes redundant. To improve the user experience, the app was modified and now only presents a host and join button. Pressing join automatically puts the user into the room hosted on the same Wi-Fi. This should provide a seamless connection phase as the user only has to press a single button, and pressing the join button when there is no host displays a message saying so. Entering host mode sends metadata needed to connect to the hosted room to the hub running the Unity editor, which is then broadcast over Wi-Fi, allowing the single-click connection for all users.

4.2.3 Usability testing

A user test was conducted to test the usability of the application, and the visualization, before any further development. Only five users are needed to identify 75-99% of all usability problems in an application [52]. Therefore, five people from the department were gathered and took part in this user test.

All the user subjects were told to open the application, walk around to detect the floor area, and join the created room. These simple instructions were all that was given, and they had not tried the app in advance. Connecting to the room worked for everyone, and no one was in doubt about how to connect. None of the subjects had problems detecting the floor, but since ARCore can detect various surface areas, some subjects got a bit confused about what was happening, which could be problematic. Four of the subjects would also like the wrongly detected areas to disappear, while the last one did not care much for it.

Multiple surfaces must be detected in case the first detected area is not the correct one, so to avoid confusion from the users, they should be told how the detection works. A possibility could be to have all detected planes vanish, except the one with the feature points that correspond to the cloud anchor. However, due to time limitations, this feature was left unfinished.

4.2.4 Interactive elements

Without interaction, the application users have to watch the visual effects through their mobile devices passively. Users have to be actively involved, or else it might get uninteresting quickly. A simple and easy way to draw attention and interest is by adding interactive elements. The possibilities of interaction with digital content are endless, and it is all about creating the best experience.
The interactive element chosen for this application was sliders that change the appearance of the visual effects. In order to preserve the essence of the original visual effect, color and behavior can not be adjusted. It should be possible to understand what is visually going on, even with sliders changing the visual’s appearance. Three sliders were added, one controlling the particle trail width, and two controlling a noise module. Noise modules add noise/turbulence to particle movement, and the sliders control how strong the overall noise effect is and its frequency. These sliders only affect particle system A and can be seen in figure 4.19. The application also logs the sliders, meaning each user’s slider values can be collected and analyzed, which could be used to find out what effects the users liked the most. Logging is controlled on the hub and can be turned on and off at any time.

![Figure 4.19: The sliders controlling the animation, seen at different values. The red slider (top) determines the particle trail width/ratio. The green slider (middle) controls noise strength. The purple slider (bottom) controls the noise frequency.](image)

### 4.3 Video demo

Figure 4.20 shows the system in use recorded by three different mobile devices, one being the host, and the others have joined the session. Some minute differences can be seen in the visuals since the animations are fully rendered on the device. Each device tries to update the game logic as fast as possible, meaning the tick rate of each device could differ. Rapid changes in the data sent from the hub then result in slight visual variations from device to device, as the received data which the visuals are generated from and the game logic updates happen independently.
4.4 Experiment planning

This section covers the design process and the goals for the chosen experiment, and an initial experiment that had to be discarded.

Before creating the setup for the experiments, the research questions that were going to be answered were defined:

- Can AR improve a live performance?
  
  Improving in this context would mean that the audience perceived the performance as more fun, enjoyable, or exciting with the addition of AR than viewing the exact same performance without AR.

- Can AR improve the perception of muscle intensities?
  
  In this context, improved perception of muscle intensities is defined as an increase in the ability to correctly classify muscle intensities in means of strength.

4.4.1 Original experiment

This is the original experiment that was planned, but due to restrictions and circumstances related to the COVID-19 pandemic, this experiment could not be carried through. In particular, gathering an audience would not have been possible. The lab was also inaccessible most of the time, halting experiments using the motion capture system. Further development and testing of interactive elements were also discontinued, as no audience would be using the application.
The application was created with the intention to be used by an audience. Therefore we want to gather as many people as possible with android devices. Since this would take place in the motion capture lab, we decided on a maximum of 20 audience members, as this is probably the limit of people that can comfortably fit inside the lab. A separate Wi-Fi was also set up in the lab to ensure stability for many simultaneous users. The participants would then stand in a circle around the motion capture area with a performer in the middle. This gives each audience member a unique viewing angle, but with the possibility to move around a bit.

In the middle, the performer would wear the motion-tracker armband and the Myo armband, and play music generated from EMG data (*RAW: A Muscle-based instrument*)\(^7\) ([1]). First one performance with just the muscle-based instrument, then another one where everyone uses the AR application. The idea behind this experiment is that the sound created from the instrument could be confusing if the audience can not correctly interpret the hand movements and muscle intensity, as some gestures can be very subtle. Watching the same performance with AR could then possibly improve the perceived understanding of muscle intensity and gestures, making it a more pleasant experience. This could also answer if AR can improve a live performance. At the end of the performance, the audience would have given feedback in the form of a survey. The preparation of this survey started just before the experiment plans changed. Therefore, no details of this survey are shown.

### 4.4.2 AR only experiment

Due to the circumstances, the motion capture lab was also no longer available, and another experiment quickly had to be planned. In order to get feedback from users, the experiments had to be entirely online-based. Testing if AR could improve a live performance was deemed to be no longer possible, but testing if AR can enhance the perception of muscle intensities was still possible.

Some modifications were made to the already existing software, making it independent from the motion capture data. These changes made it possible to move the visual effects around the room and adjust the height, and could then be manually lined up with a person’s arm. The existing visuals could be used, but the adjustable effects using the sliders, previously seen in figure [4.19](https://youtu.be/_dzA5pl9k), were left unused as participants would not be using the application. The modified app was then used to record the visual effects so it could be used in the experiments. Sharing images and videos from the app would then make it possible still to test the user’s interpretation of the visuals.

---

\(^7\) Videolink: [https://youtu.be/_dzA5pl9k](https://youtu.be/_dzA5pl9k)
Chapter 5

Experiments and results

This chapter covers the user studies, results, and analysis. Two experiments were conducted, a primary user study experiment, and a secondary user study that builds on the primary experiment.

5.1 Primary experiment

The experiment was designed in a way that allowed users to participate from home. Therefore, the experiment is mostly tasks and questions related to images and pre-recorded videos in the form of a questionnaire. The subject’s main task is to identify high muscle intensities, to test their perception of muscle intensities. The questionnaire was shared online, and a total of 37 individuals participated.

5.1.1 The questionnaire

This section shows the tasks and questions in the order presented in the questionnaire. All participants answered the questionnaire online with no time limit. Additionally, tasks are thoroughly explained to avoid confusion. As a prerequisite, a picture of the Myo armband was shown together with the following text:

This is a Myo armband. The armband contains eight sensors that wirelessly measure the electrical activity in your forearm muscles. So what does this mean? Resting your arm would register as low activity while clenching a fist would produce high activity. It is just not the overall activity that is measured, but the different muscles used. Clenching a fist and spreading your fingers might both produce high overall activity, but these hand gestures use different muscles. The sensors register this, making various hand gestures distinguishable.

The first five tasks are to select the image which you think has the highest muscle intensity.
Task 1

In Figure 5.1, which of the following do you think produces the highest muscle activity?

Additional description: At first glance, it might look like identical images, but muscle activity does vary in all the photos. Try to look for subtle differences. If you can’t decide which fist is clenching the hardest, take a guess.

Task 2

In Figure 5.2, which of the following do you think produces the highest muscle activity?

Additional description: Exactly the same as in task 1, but this time the photo is from a different angle and further away.

Task 3

In Figure 5.3, which of the following do you think represents the highest muscle activity?

Additional description: In the following image, augmented reality has been used to visualize the sensor data from the armband. The visualization is dynamically created through a smartphone video app,
meaning the visual effects are not manually edited, but generated from
the sensor data in real-time.
Image 3 shows clenching fists just like in Image 2, but with added
visual effects generated from the muscle activity. Without telling you
what the different aspects of the visual effects mean, which visual
effect do you think is generated from the fist with the highest muscle
activity? The levels of muscle intensity here are identical to the ones
in Image 2 (in random order).

Task 4

In Figure 5.4, which of the following do you think produces the highest
muscle activity?

Additional description: Same as previous tasks, but this time with
spreading fingers.

Task 5

In Figure 5.5, which of the following do you think represents the highest
muscle activity?

Additional description: Using the same visualization as before,
which visual effect do you think is generated from the spreading
fingers with the highest muscle activity?
Question 1
Do you feel that the visual effects made it easier to distinguish the different muscle intensities?
[Yes / No]
[Why / why not?]

Task 6
Which visual effect clip do you think corresponds to the following hand gesture clips? (Figure 5.6)

Additional description: The following video shows eight clips. Clips 1, 2, 3, 4 show visual effects. Clips A, B, C, D show hand gestures. The task is to match the hand gestures with the visual effects. Each numbered clip corresponds to an alphabetized clip. The orders have been randomized, meaning the hand gesture right below a visual effect might not be the correct one.
Give your best effort, but if you can’t decide, take a guess. You can watch the video as many times as you like, pause it, fast forward, etc.

Question 2
Do you have any previous experience that you think might have affected your ability to solve these tasks?
[Yes / Not sure / No]

Question 3
How well do you think the visual effects represent muscle activity?
[Rate 1-5.]
1 = poorly, 5 = very well.

Question 4
Is there anything particular you liked or disliked about the visual effects?
Figure 5.6: A frame from the questionnaire video. Videolink: [https://youtu.be/0iLUERyyXPE](https://youtu.be/0iLUERyyXPE)

Question 5

What do you think the color of the visual effects represents?

Question 6

Any other comments you would like to add? (optional)

5.2 Primary experiment results

This section presents the primary experiment results. The questionnaire was open online for two weeks, and no demographic information was gathered from the 37 participants. However, through verbal confirmation, it is known that almost half of the responses came from students at the Department of Informatics. The total outputted intensity value from the Myo armband determined the correct answers for the first five tasks. The maximum intensity value in each task does vary slightly, but was not considered an issue because the task is to identify the highest level of intensity compared to the others, not identify a specific intensity.

5.2.1 Fist gesture

Tasks 1-3 all show a clenching fist gesture, two of the tasks using images taken by a regular camera, and one with AR. The results for these tasks
are shown in figures 5.7, 5.8, and 5.9. With five options for each task, the chance of randomly guessing the correct answer is 20%.

- Task 1 shows close-up photos of a fist, and the results show that 7 subjects (19%) chose the correct answer.
- Task 2 shows photos taken from a distance (approximately 2m) of a fist. A couple more subjects got this task correctly, with a total of 9 correct answers (24%).
- Task 3 also shows photos taken from a distance, but with the AR effects. In this task, a clear majority managed to identify the highest muscle intensity, with 29 correct answers (78%).

Figure 5.7: Task 1 (Fist close-up) distribution of votes. The red bar indicated the correct image.

Figure 5.8: Task 2 (Fist distant) distribution of votes. The red bar indicated the correct image.

Figure 5.9: Task 3 (Fist AR) distribution of votes. The red bar indicated the correct image.

Ordering the answers from tasks 1-3 into the levels of intensity (table 5.1), we see that a majority of the subjects voted for the higher level intensities for all tasks. If both the highest and second-highest intensity were accepted as correct, the results change drastically.

- In task 1, 29 subjects (78%) selected a high intensity.
• In task 2, the photos from a distance, 21 subjects (57%) selected a high intensity.

• In task 3, with the AR effects, 35 subjects (95%) selected a high intensity.

<table>
<thead>
<tr>
<th>Task</th>
<th>Low</th>
<th>Mid-low</th>
<th>Mid</th>
<th>Mid-high</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Fist close-up)</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>22</td>
<td>7</td>
</tr>
<tr>
<td>2 (Fist distant)</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>3 (Fist AR)</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>29</td>
</tr>
</tbody>
</table>

Table 5.1: Task 1-3. The number of votes in order of intensity, from low to high.

5.2.2 Fingers spread gesture

Tasks 4 and 5 shows a spreading fingers gesture. The results for these tasks are shown in figures 5.10 and 5.11. Again, the chance of randomly guessing the correct answer is 20%.

• Task 4 shows photos taken from a distance of spreading fingers, and the results show that 6 subjects (16%) chose the correct answer.

• Task 5 shows similar photos, but with the AR effects. 28 subjects (76%) chose the correct answer, almost the same as in task 3.

Figure 5.10: Task 4 (Fingers spread) distribution of votes. The red bar indicated the correct image.

Figure 5.11: Task 5 (Fingers spread AR) distribution of votes. The red bar indicated the correct image.

Ordering the answers from these two tasks into the levels of intensity (table 5.2), we see that a majority of the subjects voted for the higher level intensities for both tasks. Again, if both the highest and second-highest intensity were accepted as correct, the results look quite different.

• In task 4, 24 subjects (65%) selected a high intensity.
• In task 5, with the AR effects, 31 subjects (84%) selected a high intensity.

<table>
<thead>
<tr>
<th>Task</th>
<th>Low</th>
<th>Mid-low</th>
<th>Mid</th>
<th>Mid-high</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (Fingers spread)</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>5 (Fingers spread AR)</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 5.2: Task 4 & 5. The number of votes in order of intensity, from low to high.

5.2.3 Matching pairs

In task 6, the subjects were told to match videos of hand gestures with the corresponding visual effect. No information about how the visuals were designed was given in the questionnaire, meaning the answers are based on their own interpretation. This task was created to see if the effects are perceived as reasonable and logical to the subjects. If the visuals do not make sense, it could cause confusion rather than enjoyment.

The results for task 6 are shown in figure 5.12.

- For clip A, 25 subjects (68%) matched the correct visuals.
- For clip B and C, 21 subjects (57%) chose the correct visuals.
- For clip D, 22 subjects (59%) matched correctly.
• A total of 17 subjects (46%) got all pairs correct.
• 6 subjects (16%) got two pairs correct in total.
• 9 subjects (24%) only got one pair correct.
• 5 subjects (14%) mismatched all the pairs.

5.2.4 The questions

Question 1

Shown in figure 5.13, the results from question 1 reveals that 33 subjects (89%) think that the visual effects made it easier to distinguish the intensities. Some of the reasonings are quoted below:

"The images are very similar. Visualizers have more distinct differences."

"Larger ‘bursts’ seemed to indicate higher intensity."

"Due to my experience with games, more circles and stronger colors (closer to red) often mean ‘more energy’ or similar. All the images without the visualization were basically guesses."

"It was impossible without the effects. The effects were clear and communicating."

"Entrained in my mind is the meaning of visual effects in video games, and there is a vaguely defined ‘language’ of what colors and effects mean what. Your effects follow these rules, thus letting me understand the muscle activity."

"Both yes and no, if the colors are explained, yes. A bit confused on the beginning whether it was different muscles active or on an intensity scale where purple was not active and red very active."

Question 2

The results from question 2 (figure 5.14) show that a majority of the subjects (60%) did not have any previous experience. The conditions for what qualifies as prior knowledge was not stated in the questionnaire. Therefore, the responses are based on what they themselves consider as previous experience.
Figure 5.13: Question 1 results. *Do you feel that the visual effects made it easier to distinguish the different muscle intensities?*

Figure 5.14: Question 2 results. *Do you have any previous experience that you think might have affected your ability to solve these tasks?*

**Question 3**

Question 3 results (5.13) shows the Likert scale distribution of the responses, with 3 being neutral. No subjects selected any lower than 3, and a majority (70%) decided on 4. The mean value of the responses is 3.97, and the median value is 4.

Figure 5.15: Question 3 results. *How well do you think the visual effects represent muscle activity?* 1 = poorly, 5 = very well.
Question 4

Is there anything particular you liked or disliked about the visual effects?

Most of the responses to this question were that they liked the visual effects in general, especially the colors. Some of the more noteworthy responses are listed below:

"I feel the visualization is very intuitive."

"The color differences are clear, which when given explicit meaning, will be clear to understand."

"I liked them in general, but it was a bit hard to differentiate between medium activity and hard activity."

"I liked seeing a clear change in the animations when the hand moved. Maybe having it change even more would help."

"The various colors are not obvious, this should be clarified with some kind of scale."

"The colors are nice, and the way the visualization changes when the hand moves are very beautiful to look at."

"I'm severely colourblind and had no trouble with distinguishing the colours."

Question 5

What do you think the color of the visual effects represents?

No subjects were told what any aspects of the visualization mean. This was planned to test if the visual effects are intuitive enough on their own. There is a lot of variation in the responses, but by grouping it into categories, we see a lot of similarities in the answers. The list below shows the most frequent categories in descending order, with colors indicating the intensity as the most popular answer.

1. Warmer colors mean higher intensity.
2. The different muscles being used.
4. How clenched the fist is.
5.3 Primary experiment analysis

This section shows a statistical analysis of the tasks and quantitative questions, and an analysis of the qualitative questions.

5.3.1 Fist gesture analysis

Selecting the highest intensity

When the subjects selected one of the options, there was a 20% chance of guessing the correct answer. One could expect 29.6 incorrect and 7.4 correct answers out of the 37 total responses following a uniform distribution when grouping the responses into either correct or incorrect responses. Using a chi-square goodness of fit test, the observed data can be tested against this expected distribution. A null hypothesis and an alternative hypothesis was formed to test this:

\[ H_0: \text{The grouped responses follow a uniform distribution.} \]
\[ H_A: \text{The grouped responses do not follow a uniform distribution.} \]

First testing the observed responses (table 5.3) from Task 1 (Fist close-up), with (row-1)(col-1) = 1 degree of freedom: \( X^2 = 0.026 \), and the corresponding p-value is 0.872. The test result is not significant at \( p < 0.05 \), and the null hypothesis is not rejected. When it comes to selecting the highest muscle intensity from close-up images, there is not sufficient evidence to support a conclusion that the grouped responses do not follow a uniform distribution.

\[
\begin{array}{|c|c|}
\hline
\text{Incorrect} & \text{Correct} \\
\hline
\text{Observed} & 30 & 7 \\
\text{Expected} & 29.6 & 7.4 \\
\hline
\end{array}
\]

Table 5.3: Task 1 observed vs expected values

Next, testing the observed responses (table 5.4) from Task 2 (Fist distant), with 1 degree of freedom: \( X^2 = 0.432 \), and the corresponding p-value is 0.511. The test result is not significant at \( p < 0.05 \), and the null hypothesis is not rejected. When it comes to selecting the highest muscle intensity from photos taken from a distance, there is not sufficient evidence to support a conclusion that the grouped responses do not follow a uniform distribution.

\[
\begin{array}{|c|c|}
\hline
\text{Incorrect} & \text{Correct} \\
\hline
\text{Observed} & 28 & 9 \\
\text{Expected} & 29.6 & 7.4 \\
\hline
\end{array}
\]

Table 5.4: Task 2 observed vs expected values

Performing the same test with the observed responses (table 5.5) from Task 3 (Fist AR), with 1 degree of freedom: \( X^2 = 78.808 \), and the
corresponding p-value is < 0.00001. The test result is significant at p < 0.05, and the null hypothesis is rejected, and thus the alternative hypothesis is accepted. It is possible to conclude that correctly selecting the highest muscle intensity from AR images is statically significant compared to guessing. Since both the non-AR tasks were not statistically significant, it is also possible to conclude that AR makes it easier to select the highest muscle intensity when looking at a clenched fist.

<table>
<thead>
<tr>
<th>Incorrect</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>8</td>
</tr>
<tr>
<td>Expected</td>
<td>29.6</td>
</tr>
</tbody>
</table>

Table 5.5: Task 3 observed vs expected values

Selecting a high intensity

As we saw in the results section, for both the non-AR images, the second-highest intensity got the most votes. By counting both the highest and the second-highest as correct answers, the chance of randomly guessing the correct answer would be 40%. Using this, one could expect 14.8 correct and 22.2 incorrect answers from the responses from the total 37. Performing the same test with the same null hypothesis and alternative hypothesis, we see some interesting results shown in table 5.6. All three tests are significant at p < 0.05, and the null hypothesis is rejected. It is possible to conclude that the ability to select a high muscle intensity from the images is in all three cases statically significant compared to guessing.

<table>
<thead>
<tr>
<th>Incorrect</th>
<th>Correct</th>
<th>X²</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fist close-up</td>
<td>8</td>
<td>29</td>
<td>22.707 &lt;0.00001</td>
</tr>
<tr>
<td>Fist distant</td>
<td>16</td>
<td>21</td>
<td>4.328 0.037</td>
</tr>
<tr>
<td>Fist AR</td>
<td>2</td>
<td>35</td>
<td>45.950 &lt;0.00001</td>
</tr>
<tr>
<td>Expected</td>
<td>22.2</td>
<td>14.8</td>
<td></td>
</tr>
<tr>
<td>P(Expected)</td>
<td>0.6</td>
<td>0.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6: Chi-square and p-values for tasks 1-3 when accepting the second highest intensity as correct.

Following, it is desired to test if there is any significant difference between using AR or not. To test this, a null hypothesis and an alternative hypothesis was formed:

\[ H_0: \text{Subjects looking at AR images are equal or worse at identifying a high muscle intensity than without AR.} \]

\[ H_A: \text{Subjects looking at AR images are better at identifying a high muscle intensity than without AR.} \]

Using a z-score test for two population proportions on Fist AR and Fist close-up gives \( z = 2.04 \), corresponding to the one-sided p-value of 0.021. The
result is significant at p < 0.05, and the null hypothesis is rejected. Using the same test on Fist AR and Fist distant gives z = 3.793, corresponding to the one-sided p-value of 0.00008. The result is significant at p < 0.05, and the null hypothesis is rejected. It is possible to conclude that AR makes it easier to spot a high muscle intensity when clenching a fist.

5.3.2 Fingers spread gesture analysis

Selecting the highest intensity

Again, one could expect 29.6 incorrect and 7.4 correct answers out of the 37 total responses following a uniform distribution when grouping the responses into either correct or incorrect responses. Using the chi-square goodness of fit test, the observed data can be tested against this expected distribution. The null hypothesis and an alternative are the same as in the previous goodness of fit test.

Testing the observed responses (table 5.7) from Task 4 (Fingers spread), with 1 degree of freedom: \(X^2 = 0.331\), and the corresponding p-value is 0.565. The test result is not significant at p < 0.05, and the null hypothesis is not rejected. When it comes to selecting the highest muscle intensity from images of spreading fingers, there is not sufficient evidence to support a conclusion that the grouped responses do not follow a uniform distribution.

<table>
<thead>
<tr>
<th>Incorrect</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>31</td>
</tr>
<tr>
<td>Expected</td>
<td>29.6</td>
</tr>
</tbody>
</table>

Table 5.7: Task 4 observed vs expected values

Further, testing the observed responses (table 5.8) from Task 5 (Fingers spread AR), with 1 degree of freedom: \(X^2 = 71.682\), and the corresponding p-value is < 0.00001. The test result is significant at p < 0.05, and the null hypothesis is rejected. It is possible to conclude that selecting the highest muscle intensity from AR images of spreading fingers is statically significant compared to guessing. It is also possible to conclude that AR makes it easier to select the highest muscle intensity when looking at spreading fingers.

<table>
<thead>
<tr>
<th>Incorrect</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>9</td>
</tr>
<tr>
<td>Expected</td>
<td>29.6</td>
</tr>
</tbody>
</table>

Table 5.8: Task 5 observed vs expected values
Selecting a high intensity

Interestingly, in this case as well, the image with the second-highest intensity got the most votes for the non-AR images. If the requirements for correct were changed to selecting one of the two highest intensities, the chance of randomly guessing the correct answer would be 40%. Performing the same test with the same null hypothesis and alternative hypothesis, gives the p-values shown in table 5.9. In both cases, the tests are significant at p < 0.05, and the null hypothesis is rejected. It is possible to conclude that the ability to select a high muscle intensity from the images in both cases is statically significant compared to guessing.

<table>
<thead>
<tr>
<th></th>
<th>Incorrect</th>
<th>Correct</th>
<th>X²</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingers spread</td>
<td>13</td>
<td>24</td>
<td>9.531</td>
<td>0.002</td>
</tr>
<tr>
<td>Fingers spread AR</td>
<td>6</td>
<td>31</td>
<td>29.553</td>
<td>&lt;0.00001</td>
</tr>
<tr>
<td>Expected</td>
<td>22.2</td>
<td>14.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PExpected</td>
<td>0.6</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.9: Chi-square and p-values for tasks 4 & 5 when accepting the second highest intensity as correct.

Next, it is desired to test if there is any significant difference between using AR or not for spreading fingers. To test this, the same null and alternative hypothesis used in the previous z-test is used. A z-test on Fingers spread and Fingers spread AR gives a z score of 1.8628, and a corresponding one-sided p-value of 0.031. The result is significant at p < 0.05. It is possible to conclude that AR makes it easier to select a high muscle intensity when looking at spreading fingers. Since there is a statistical difference for both hand gestures viewed in AR, the conclusion is that AR can make it easier to recognize high levels of muscle intensities.

5.3.3 Matching pairs analysis

In the matching pairs task, random guesses have to be taken into account. There are four video pairs, giving a total of 4! = 24 possible permutations of pairs. Only one of these permutations have all pairs right, resulting in a probability of 1/24 to guess all correctly. Table 5.10 shows the probability distribution for all the possible numbers of correct pairs.

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(n)</td>
<td>10/24</td>
<td>8/24</td>
<td>5/24</td>
<td>1/24</td>
</tr>
</tbody>
</table>

Table 5.10: The probabilities of getting exactly n number of correct pairs. Note that n=3 is missing as it is impossible to get exactly three pairs correct.

Next, the observed responses must be tested against the probabilities of random guesses, with the following null hypothesis and alternative hypothesis:

H₀: The responses follow the probability distribution in table 5.10.
H_A: The responses do not follow the probability distribution in table 5.10.

To test if the observed data follows this expected distribution, the chi-square goodness of fit test is used. Testing with the observed responses shown in table 5.11 with 3 degrees of freedom: $X^2 = 163.93$, and the corresponding p-value is $< 0.00001$. The test result is significant at $p < 0.05$, and the null hypothesis is rejected, and thus the alternative hypothesis is accepted. It is possible to conclude that the subject responses are significantly different from random guesses.

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>Expected</td>
<td>15.4</td>
<td>12.3</td>
<td>7.7</td>
<td>1.5</td>
</tr>
<tr>
<td>P(n)</td>
<td>10/24</td>
<td>8/24</td>
<td>5/24</td>
<td>1/24</td>
</tr>
</tbody>
</table>

Table 5.11: Task 6 observed vs expected values. n is the number of correct pairs.

This task was created to see if the visual effects were expressive and intuitive to the subjects. 17 (46%) of the subjects identified all the correct pairs, but might not be enough to call the visual effects intuitive. It is, however, possible to conclude the AR effects are somewhat meaningful, as almost half of the subjects got it correct. It is also worth noting that some subjects found this task to be difficult due to its setup:

"I found the video matching task with 4(!) options too difficult. Would have liked better if it was a series of A / B tests, where I only had to arrange two-and-two items." (Translated)

5.3.4 Question analysis

Using the answers from questions 1 and 2, it is possible to test if any previous experience makes the user feel it is easier to distinguish the different muscle intensities, with the following null hypothesis and alternative hypothesis:

H_0: Previous experience does not make you feel it is easier to distinguish the different muscle intensities.

H_A: Previous experience makes you feel it is easier to distinguish the different muscle intensities.

Applying the chi-squared test on the observed data in table 5.12 with 2 degrees of freedom, gives $X^2 = 2.912$. The p-value is 0.233. The result is not significant at $p < 0.05$, and the null hypothesis is not rejected. There is not sufficient evidence to support a conclusion that previous experience makes you feel it is easier to distinguish the different muscle intensities.

Combining the responses from questions 2 and 3, it is also possible to test if previous experience makes you think the visual effects are better at
representing muscle activity. For this, a null and alternative hypothesis was formed:

\( H_0 \): Having previous experience does not impact how you rate the visual effects.

\( H_A \): Having previous experience does impact how you rate the visual effects.

For this test, Yes and Not sure are grouped since it is desirable to test for those who clearly have no previous experience. The Likert scale distribution for previous experience is shown in table 5.13 and the two groups are compared using a nonparametric test for two-groups (Mann-Whitney U test). This results in a z-score of -1.655, and the p-value is 0.099 using a two-sided test. The result is not significant at \( p < 0.05 \). The null hypothesis is not rejected, and there is not sufficient evidence to support a conclusion that having previous experience impacts how you rate the visual effects.

<table>
<thead>
<tr>
<th>Likert rating</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes/Not sure</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>No</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>14</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.13: The number of votes on the Likert scale distributed on previous experience.

These two last tests show that previous experience does not affect your opinion on the AR effects, but it is also interesting to see how the subjects performed with previous experience as the factor. From the results, 17 subjects correctly matched all the pairs in tasks 6. If those 17 are compared against the 20 that did not get all correct, we see in table 5.14 that the distribution is quite similar. To test if previous experience has any significance, the following null hypothesis and alternative hypothesis was formed:

\( H_0 \): Having previous experience does not impact your ability to solve the tasks.

\( H_A \): Having previous experience impacts your ability to solve the tasks.
A chi-squared test with 2 degrees of freedom gives $X^2 = 0.359$. The corresponding p-value is 0.836, and the result is not significant at $p < 0.05$. The null hypothesis is not rejected, and there is not sufficient evidence to support a conclusion that having previous experience impacts your ability to solve the tasks. From these three statistical analyzes involving previous experience, it is possible to conclude that previous experience is not a factor when it comes to how the subjects perceive the visuals.

### The qualitative questions

From the responses in question 4, presented in the results section, the main concern from the subjects was if they interpreted the visuals correctly. This was somewhat expected, as no explanation was given. Those who felt a bit confused wanted extra information so they could know if they interpreted it correctly. Some reported the visualization was intuitive, but in question 5 indicated they misinterpreted the visuals.

A significant portion of the subjects thought that warm colors indicated high intensity. This misunderstanding could be a result of the clenching fist gesture showing red-colored effects, and since it is the gesture demonstrated with the most upper strength, this would seem logical to them. The red effects when clenching a fist were just a coincidence determined by the Myo armband’s placement on the forearm. This could explain why some rated the visuals a 5 but answered incorrectly on some tasks.
5.4 Secondary experiment

This section shows the second experiment that was conducted, with results and analysis. The format for this experiment is similar to the primary one and is also entirely online.

From the abandoned original experiment, the sonification from Erdem and Jensenius 2020 [1] was utilized in this experiment. Since it could not be used in an audience setting, it was added as sound to videos instead. These sounds were not made to represent muscle intensity directly, as the AR effects are. Therefore, an intensity test like in the first experiment would be an unfair comparison. The sonification and AR effects are however, both generated from the Myo armband signals. Using the same video format as in task 6 (matching pairs, primary experiment), the sonification can be used to test if sound makes more sense than visuals for recognizing hand movements. Naturally, since the videos had to contain sounds, not all clips could be played at once, like in task 6. Hence, the video format is slightly different, but the concept is the same.

**The sonification:** Incoming raw EMG signals are first normalized and then written recurrently into buffers at every 50 samples (250ms). This provides two dynamic wavetables that are continuously updated. Then, the wavetables are brought to an audible range, and time-scaled sawtooth signals control their frequency spectra.

5.4.1 The questionnaire

This section shows the tasks and questions in the order presented in the questionnaire. All subjects participated online with no time limit, and the same prerequisite about the Myo armband as in the primary experiment questionnaire was given.

**Task 1**

Which hand gesture clip do you think corresponds to the following audio clips? (Figure 5.16)

Additional description: The following video shows four numbered video clips that show hand gestures. These video clips are looped four times, but with a different audio clip playing each time. The audio clips are alphabetized and displayed under the video clips. The hand gestures in video clips 1,2,3,4 generated the audio clips A, B, C, D. Your task is to match the hand gestures with the audio clips. Give your best effort, but if you can’t decide, take a guess. You can watch the video as many times as you like, pause it, fast forward, etc.
Figure 5.16: A frame from the video in task 1. Videolink: https://youtu.be/125HZOgHtv4

Task 2
Which hand gesture clip do you think corresponds to the following visual effect clips? (Figure 5.17)

Additional description: The next video is similar to the first one, but this time a visual effect video clip is playing together with the sound. These visual effects were also generated live from the hand gestures. The task is exactly the same, match the numbered clips with the alphabetized clips.

Question 1
Which of the tasks do you felt was the easiest?
[Sound / Sound and visuals / Equal difficulty]

Question 2
Do you have any previous experience that you think might have affected your ability to solve these tasks?
[Yes / Not sure / No]

Question 3
Any comments you would like to add? (optional)
5.5 Secondary experiment results

This section presents the secondary experiment results. The questionnaire was open online for one week, and 23 subjects participated. Again, no demographic information was gathered, but participants from the first experiment were notified about this second questionnaire.

5.5.1 Sound

Figure 5.18 shows the results for task 1, where the subjects were told to match hand gestures with sounds. No information about the sonification designed was given, meaning the answers are based on their own interpretation of the sounds.

- For gesture 1 and 3, 10 subjects (43%) chose the correct sound clips.
- For gesture 2, 13 subjects (57%) chose the correct sound clip.
- For gesture 4, 14 subjects (61%) chose the correct sound clip.
- 7 subjects (30%) got all pairs correct.
- 5 subjects (22%) got two pairs correct in total.
- 9 subjects (39%) only got one pair correct.
Figure 5.18: Task 1 distribution of votes per clip. The checkmark indicates the correct pair.

- 2 subjects (9%) mismatched all the pairs.

### 5.5.2 Sound and visuals

In task 2, the subjects were told to do the same task again, but this time with additional visuals. The results for task 2 is shown in figure 5.19.

Figure 5.19: Task 2 distribution of votes per clip. The checkmark indicates the correct pair.

- For gesture 1 and 4, 19 subjects (83%) chose the correct video clips.
- For gesture 2 and 3, 17 subjects (74%) chose the correct video clips.
- 16 subjects (69%) got all pairs correct.
- 3 subjects (13%) got two pairs correct in total.
- 2 subjects (9%) only got one pair correct.
• 2 subjects (9%) mismatched all the pairs.

• 12 subjects (52%) got more correct pairs in task 2 than in task 1.

• 2 subjects (9%) got fewer correct pairs in task 2 than in task 1.

• 9 subjects (39%) got the same amount of correct pairs in both tasks.

• 7 subjects (30%) got all pairs correct in tasks 1 and 2.

Question 1

Figure 5.20 shows the results from question 1, and equal difficulty got the most votes. Only one of the seven who got all pairs correct on both tasks voted for sound and visuals. Some of the comments to this question are listed below:

"Felt like the sound was a bigger factor than the visualization. If the tests were with only one component(sound or visualization) the test with sound would be easier to solve."

"The tasks seemed equally hard as sound was an important factor and was included in both. If the tests were done with only one source(only sound or visualization) the test with only sound would be far easier."

"I think it is easy when you compare colors, effort, and tension in the sound."

Figure 5.20: Question 1 results. Which of the tasks do you felt was the easiest?

Question 2

The results from question 2 (figure 5.21) show that a majority of the subjects (57%) did not have any previous experience. This distribution is almost identical to the distribution of responses to the same question in the primary experiment (figure 5.14).
Figure 5.21: Question 2 results. *Do you have any previous experience that you think might have affected your ability to solve these tasks?*

- 7 of the 8 subjects (88%) who voted for *sound and visuals* in question 1 responded with no previous experience in question 2.
- 5 of the 7 subjects (71%) who reported having previous experience voted for *equal difficulty*. One voted for *sound and visuals*, and the last one voted for *sound*.

### 5.6 Secondary experiment analysis

This section shows a statistical analysis of the tasks and questions in this second experiment, and a comparison of the results in both experiments at the end.

#### 5.6.1 Video analysis

When analyzing the video tasks, random guesses have to be taken into account. These videos have the same number of pairs as the video task in the primary experiment, giving the probabilities that were previously shown in table 5.10. The observed data can be tested against the distribution of random guesses using the following null hypothesis and alternative hypothesis:

- **$H_0$:** The responses follow the probability distribution in table 5.10.
- **$H_A$:** The responses do not follow the probability distribution in table 5.10.

Using the chi-square goodness of fit test, the observed data can be tested against the expected distribution. Testing with the observed responses shown in table 5.15 with 3 degrees of freedom:

- **Task 1 (sound):** $X^2 = 47.59$, and the corresponding p-value is < 0.00001. The test result is significant at $p < 0.05$, and the null hypothesis is rejected, and thus the alternative hypothesis is accepted.
- **Task 2 (sound and visuals):** $X^2 = 264.25$, and the corresponding p-value is < 0.00001. The test result is significant at $p < 0.05$, and the null hypothesis is rejected, and thus the alternative hypothesis is accepted.

![Bar chart showing results of Question 2](chart.png)
Table 5.15: Tasks 1 and 2, observed vs expected values. n is the number of correct pairs.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed (sound)</td>
<td>2</td>
<td>9</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Observed (sound and visuals)</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>Expected</td>
<td>9.6</td>
<td>7.7</td>
<td>4.8</td>
<td>0.9</td>
</tr>
<tr>
<td>P(n)</td>
<td>10/24</td>
<td>8/24</td>
<td>5/24</td>
<td>1/24</td>
</tr>
</tbody>
</table>

It is possible to conclude that, for both tasks, the subject responses are significantly different from random guesses.

Next, it is desirable to compare the results from the two tasks. Using the contingency table (table 5.16), it is possible to perform a chi-squared test to see if there any difference between the two observations, with the following null hypothesis and alternative hypothesis:

\[ H_0: \text{Adding visuals effects to a sonification does not improve the ability to identify the corresponding hand gestures.} \]

\[ H_A: \text{Adding visuals effects to a sonification improves the ability to identify the corresponding hand gestures.} \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound</td>
<td>2</td>
<td>9</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>Sound and visuals</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>Total</td>
<td>4</td>
<td>11</td>
<td>8</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 5.16: Contingency table for tasks 1 and 2 observations.

A chi-squared test with 3 degrees of freedom, gives \( X^2 = 8.476 \). The p-value is 0.037. The result is significant at \( p < 0.05 \), and the null hypothesis is rejected. It is possible to conclude that adding visuals effects to a sonification enhances the ability to identify the corresponding hand gestures.

5.6.2 Question analysis

In the results, we saw that the votes for the easiest task varied based on previous experience. Grouping the answers for Yes and Not sure, gives the distribution in table 5.17. To test if there is any significant difference based on previous experience, the following null hypothesis and alternative hypothesis was formed:

\[ H_0: \text{Having previous experience does not impact your opinion on which task is the easiest.} \]
\textbf{H}_A: Having previous experience impacts your opinion on which task is the easiest.

<table>
<thead>
<tr>
<th>Equal difficulty</th>
<th>Sound</th>
<th>Sound and visuals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes/Not sure</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>No</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5.17: Distribution of votes for previous experience and difficulty.

A chi-squared test with 2 degrees of freedom gives $X^2 = 6.011$. The corresponding p-value is 0.049, and the result is significant at $p < 0.05$. The null hypothesis is rejected, and thus the alternative hypothesis is accepted. It is possible to conclude that previous experience is a factor in how the subjects perceive the difficulty of the tasks. Subjects with experience think that the tasks are equally difficult, while subjects with no experience perceived the task with both sound and visuals as the easiest.

5.6.3 Comparing with the primary experiment

With the results from the primary experiment, the pair-matching using sonification can be compared with the pair-matching using only visuals. The objective for these tasks was the same, but with some differences. The pair-matching in the first experiment had sequences of different gestures, while the pair-matching in this experiment had the same repeated gesture. Comparing the observations might not give a conclusive result, but a comparison could still be interesting using the following null hypothesis and alternative hypothesis:

\textbf{H}_0: Subjects listening to the sonification are equally good at identifying the correct hand gestures compared to subjects viewing the visual effects.

\textbf{H}_A: Subjects listening to the sonification are not equally good at identifying the correct hand gestures compared to subjects viewing the visual effects.

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound</td>
<td>2 (9%)</td>
<td>9 (39%)</td>
<td>5 (22%)</td>
<td>7 (30%)</td>
<td>23</td>
</tr>
<tr>
<td>Visuals</td>
<td>5 (14%)</td>
<td>9 (24%)</td>
<td>6 (16%)</td>
<td>17 (46%)</td>
<td>37</td>
</tr>
<tr>
<td>Total</td>
<td>7</td>
<td>18</td>
<td>11</td>
<td>24</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 5.18: Contingency table for task 1 (secondary) and task 6 (primary).

A chi-squared test with 3 degrees of freedom gives $X^2 = 2.408$. The corresponding p-value is 0.492, and the result is not significant at $p < 0.05$. The null hypothesis is not rejected. There is not sufficient evidence to support a conclusion that there is a difference between listening to the
sonification and viewing the visual effects when it comes to identifying the correct gestures.
Chapter 6

Discussion

This chapter discusses the findings from the experiment and implementation, future work, and presents a conclusion to the results.

6.1 The questionnaires

This section evaluates the questionnaire design, and consequences possibly caused by the design choices.

Misinterpretations

The subjects were not told how the visual effects were generated, which was intentional to test if the effects are intuitive. As we saw in the results when the subjects guessed what the color meant, a lot of them misinterpreted the purpose of the colors and thought that warmer colors meant higher intensity. This misinterpretation is completely reasonable because a majority of the cases in the images and videos could follow this rule. Resting hands mostly shows purple or blue, and a clenching fist shows red and yellow. After looking over the images and videos in the questionnaires, it is noticeable that this warm color logic does not fit everywhere, and may have lead to several subject errors. This has possibly lead to inaccuracy in the results, and it is unknown if the subjects would have performed better if they were told how the visualization works. The second experiment could have suffered this same problem, as it followed the same setup and user information. Unfortunately, there was not enough time to do a third experiment to look into this.

The videos

The more specific test to see if the generated effects are intuitive and logical are the pair-matching tasks. There are many ways this could have been tested, but being limited to pre-recorded videos, it turned out to be an adequate test.

The pair-matching in the first experiment was considered difficult by some subjects, because of all eight videos playing simultaneously and hav-
ing several gestures in the same clips. There were two clips with three gestures and two clips with four gestures, and by analyzing the results further, it was found that the wrong clip with the same amount of gestures got the 2nd most votes in all cases, except clip B, where it got 3rd. This could mean that the number of gestures in a clip was a giveaway when deciding the pairs. In the second experiment, there were no comments about the pair-matching being difficult. No sound or effect clips were playing simultaneously, which could explain no reports of this being difficult.

A discovery about the second experiment was the sound quality. It was found that the audio had some quality loss after being uploaded to YouTube. The quality difference was not substantial, but noticeable. Also, some of the lower frequency sounds are a bit unclear when played by a speaker, but it could also vary depending on the speaker’s quality. Therefore, the subjects were recommended to use headphones in the questionnaire. Sound quality could be a possible weakness in the experiment, but should not have a significant impact on the results.

Bias

Avoiding biases is challenging when running experiments alone. Suggestive questions were being avoided, and since the questionnaire was online, all subjects could remain completely anonymous. However, when sharing the online study, the researcher is at risk of biasing the participants, since most of the responses are then more likely to come from friends and coworkers. It was not written in the questionnaire that the researcher made the effects, but of course, that was implied. From the Likert scale results in 5.15 none of the subjects rated the visual effects negatively, which is a reason to suspect bias in the study. There is also a possibility that subjects could have assisted the researcher by intentionally answering incorrectly on the tasks without the AR visualization to improve the results. These biases could have been avoided if the researcher’s identity was hidden, but would make it increasingly difficult to find participants. The discussed biases are just possibilities, and there is no reason to deem it was enough bias to affect the findings, since the experiment was as controlled as practically possible.

Randomization

The experiments lack randomization for each task, and probably affect the first tasks in the primary experiment. When the subjects were asked to select the image representing the highest muscle intensity, the images had the same order for everyone. The images were joined vertically, and comparing images far from each other could be more difficult than comparing to the adjacent ones. In task 1, the correct answer is the first image, and it only has an image to its right, and as we saw in the results, the fourth image got the most votes. This fourth image has images to the left and right with lower intensities, and this could possibly create an optical illusion, making the intensity seem higher than it is. There is no way to
know if this is actually the case, but it could be a plausible explanation. Ideally, the order of the images would have been randomized, but the form software’s limitations made this impossible.

6.2 The app

Unfortunately, due to the circumstances, the app never got properly tested with many simultaneous users in the motion capture room. However, there were some findings during the implementation worth discussing.

The application’s core functions are motion tracking and cloud anchors, which is the foundation for the system. There is almost no limitation to what could be built on top of this, and the visualization is only restricted by the designer and processing power. In a performance setting, the audience is also not required to wear or use any special equipment to participate. The only requirement is a device with Wi-Fi and a camera, which is a standard for mobile devices. For this thesis, the application was only created for android, but since Unity is cross-platform, running on multiple operating systems would be possible with some adjustments. The system in this thesis also only uses one motion capture armband, but it is not restricted to just one. It is possible to track several rigid bodies, but the mobile device’s processing power is most likely going to be an issue with many trackable items. Six floating-point numbers represent each rigid body transform, meaning the data sent over UDP would grow linearly with more tracked objects.

From an audience perspective, a clear constraint is the required floor detection. From experience, you usually have to be within a couple of meters range of the surface area to detect it. This means you have to walk up to the stage area, which could be problematic with a large audience. Once the floor is detected, there is no specific limit to how far away you can move, because ARCore continuously updates the pose when moving away, as seen in figure 6.1. Opening other applications, closing the device, sleep mode, etc. makes the app lose its tracking position, and the spectator then has to move around until the device sees enough feature points to make the visual effects appear again. Therefore, the application can be problematic to a sitting audience.

From a performer’s perspective, the system does not require any additional effort during a performance, with only having to wear the armbands, and staying within the motion capture area. The visual effects viewed by the audience do not interfere with the artist and allows for improvisation, giving them full control over their performance, as presented in the background chapter, is preferred by musical artists [42]. A potential problem is that the performer can’t view themselves with the effects, naturally. Having monitors facing the performer could solve this, but could also be distracting.
6.3 Visualization using augmented reality

The muscle activity visualization was tested on subjects to see if it can improve their perception of muscle intensities in the forearm. This was investigated with online questionnaires to test their understanding of muscle intensity and gestures. As seen in the previous chapter, the augmented reality visualization performed better at identifying high intensities than traditional viewing of images. Subtle differences that are hard to spot with the naked eye became more apparent with the AR effects. However, the experiment does not cover all types of muscle intensities, just the forearm muscles.

The subjects could also intuitively make sense of the visualization by just looking at hand gestures from the videos. When doing the same type of video experiment with sound, it was found that visuals were not significantly better than sound at matching gestures. But the visuals combined with sound performed clearly better than sound alone.

The experiments can be said to be somewhat simplistic because of the limited test environment and specific cases. All images and videos were taken in the same lighting conditions, with the armband on the right forearm on the same person. It is uncertain if the same results would occur in more general cases, but it is reasonable to think that the results would look similar. There are also some instances where even better performance could be assumed. Like a case where the user is wearing clothes over the
Myo armband, and the arm is partially obstructed, the AR visuals will remain the same.

Because the full system never got any audience feedback, it is impossible to conclude if AR can improve a live performance, which the first research question. It is possible to imagine that an audience would have liked a performance with AR effects better since the visual effects received good feedback. Results also show that the effects made it easier to perceive differences in muscle intensities, which should strengthen the interactivity and positively impact performances where muscle activity is essential. However, it is also plausible that an audience could dislike the system. Viewing the performance on a mobile device reduces the view to a tiny screen, and it could also be tiresome to hold the device up for more extended periods. This could be a possible deal-breaker for the audience. It would also have been essential to get feedback from the audience on the AR effects as a complement to the performance. The AR could be entertaining in itself, and the performance could be fantastic, but it would have been interesting to test if the AR effects improve the performance or act as a separate piece of entertainment that takes the attention from the actual performance.

6.4 Future work

6.4.1 Live performance

As previously explained, the system never got the opportunity to be tested in a performance setting. It remains to see if the system can improve a live performance. The visualization could be applied to a dance show, piano performance, guitar concert, or any form of entertainment where arm movements are essential. Since the application was created to be used by an audience, receiving feedback on the features is important for further development.

Expanding the tracking to a full motion capture suit could also open up possibilities for remote performances. A dancer could be inside a studio, and in another area, an audience could be watching the person’s dancing avatar through their devices. If the restrictions of being the same Wi-Fi was removed, people could also watch the performance in their own homes.

6.4.2 Applying the system to a robot

The system is versatile, and other sensors than EMG sensors could easily be applied. It would be interesting to see the system running using sensors from a robot. By adding trackers, multiple users could film the robot and see its sensor data visualized. This could be used as a visual debugging tool by examining its navigation, internal sensors, and gait, via displaying the information on and around the robot. Visualizing robot sensor data like
this could be worth exploring since it could be useful for various types of robotics development.

### 6.4.3 Further development

**Visuals**

The first thing that should be looked at is the colors of the visual effects because of the misinterpretations discussed earlier. A possible change is to have the color represent the intensity, with warm colors meaning high intensity. Then make the appearance of the visuals change dependent on which channels are the most active. This could reduce the chance of misinterpretation, based on what was learned from the feedback.

Another experiment should also be conducted, where the visualization details are explained before the subject answer any questions. It is reasonable to think that this would give even better results, but cannot be proved without testing. A third experiment did not fit into the available time and, therefore, was not conducted.

**Interaction**

More possible interactions should also be added. An early idea that was not implemented is to have modes that change the overall appearance and theme of the visuals, in addition to the sliders. The performer would then be in charge of the modes and select the ones most suited for each part of the performance. Additionally, the visual effects can interact with movement. The current effects follow the arm, but the arm movement does not affect the visuals. Type of effects like seen in figure 6.2 could result in interesting looking visuals in performances with a lot of movement, like a dance performance.

![Figure 6.2: A trail following the object’s movement, adding clarity to its trajectory.](image)

**Tracking and occlusion**

The motion capture setup limits the system since it only can be used where motion capture cameras are installed. Tracking methods that are using the camera and sensors on the user’s device would make the system more portable. Different tracking methods were explored in the implementation, but
motion capture was ultimately the best available option in this case, due to its stability and precision.

The system does not handle occlusion, and the effects can still be seen if real-world objects obstruct the arm’s view. The addition of depth detection can make the effects disappear behind objects and create a more believable and immersive scene.

6.5 Conclusion

This thesis has explored a shared augmented reality system used to visualize muscle intensity using a Myo armband. A complete system was built using motion capture and the AR platform ARCore, tracking the user’s arm, making real-time generated visual effects appear over the arm. Unfortunately, the system never got adequately tested, but it shows great potential for AR performances. The visualization part of the system was investigated, and user study results show that the system significantly improved the subjects’ ability to distinguish muscle intensities. A second user study compared the visualization to a sonification of hand gestures. Combining visuals and sounds showed an increased ability to identify the correct gestures, compared to only sounds. From these results, it is possible to conclude that AR can improve the perception of muscle intensities.

Positive feedback received from questionnaires encourages further development of shared augmented reality experiences and new types of performances. Efforts in investigating if AR can make live performances more attractive and exciting need additional work, but findings look promising. Further work could also generalize the systems, leading to similar applications being created in other fields like sports, education, and robotics.
Bibliography


[37] Dario Mazzanti et al. ‘Augmented Stage for Participatory Performances.’ In: NIME. 2014, pp. 29–34.


