



Faculty of Electrical Engineering,
Mathematics and Computer Science

3D-model annotation for Computer Aided Design in augmented reality

Max van IJsselmuiden
B.Sc. Thesis Creative Technology
July 9, 2018

Supervisor

dr. J. Zwiers
dr. R.J.F. Ordelman

Human Media Interaction
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
Zilverling 2031
P.O. Box 217
7500 AE Enschede
The Netherlands

Contents

Abstract	6
Acknowledgements	7
List of abbreviations	8
1. Introduction	9
1.1 Context.....	9
1.2 Problem statement	9
1.2.1 Research questions	11
2. State of the Art	12
2.1 Garment simulation.....	12
2.1.1 The early days.....	12
2.1.2 Recent advances.....	12
2.2 Tangible interaction in Augmented Reality	12
2.3 Feedback methods	12
2.3.1 Feedback forms	18
2.3.2 Content of feedback	18
2.3.3 Timing of feedback	19
2.3.4 Constructive feedback.....	19
2.4 Annotation in Augmented Reality	19
2.5 Hardware.....	21
2.5.2 AR versus VR.....	21
2.5.3 Chosen hardware for prototype	22
2.5.4 Background information of the Meta 2	23
2.5 Conclusion	23
3. Methods and techniques.....	25
3.1 Design processes.....	25
3.2 Methods.....	26
3.2.1 Stakeholder analysis.....	26
3.2.2 PACT analysis	26
3.2.3 Product and user requirements (MoSCoW).....	26

3.2.4 User and expert interviews.....	27
4. Ideation	28
4.1 Stakeholder analysis	28
4.1.1 Hecla Professional Audio & Video Systems	29
4.1.2 PVH.....	29
4.2 PACT analysis	30
4.3 Brainstorm	33
4.4 User Interfaces.....	33
4.5 Concept ideas.....	34
4.5.1 Summarizing voice input in a word cloud.....	34
4.5.2 Skype sessions during feedback recording	34
4.5.3 Different models for fitting.....	34
4.5.4 Grouping annotations based on location.....	35
4.5.5 Multi-user annotating.....	35
4.5.6 Filtering of annotations	35
4.5.7 ‘Step in’ interaction with the garment.....	35
4.5.8 ‘Pushpin’ selection interaction.....	35
4.5.9 Exploded view	36
4.6 Observations	36
4.6.1 General observations	36
4.6.2 Interaction observations	37
5. Specification	38
5.1 Prototype description	38
5.2 Product and user requirements (MoSCoW)	39
6. Realization	41
6.1 Prototype	41
6.1.1 First iteration.....	41
6.1.2 Second iteration.....	41
6.1.3 Final iteration.....	41
6.2 First iteration	42
6.2.1 Iteration requirements	42
6.2.3 Procedure	42

6.3 Second iteration	45
6.3.1 Iteration requirements	45
6.3.2 Procedure	45
6.4 Third iteration.....	48
6.4.1 Iteration requirements	48
6.4.2 Procedure	48
6.5 System structure	51
6.5.1 User interface	51
6.5.2 User interaction	51
6.6 Implementation	53
6.6.1 Software	53
6.6.2 Hardware	54
7. Evaluation.....	56
7.1 First prototype	56
7.2 Second prototype	57
7.3 Final prototype	58
7.4 Functional Testing.....	60
7.2 User evaluation	61
7.3 Internal evaluation	63
7.3.1 Project processes	63
7.3.2 Final prototype.....	64
8. Conclusions.....	65
8.1 State of the Art	65
8.2 Stakeholder identification and PACT analysis	65
8.3 Ideation and specification	65
8.4 Realization and evaluation	66
8.5 Research question.....	67
9. Future work	68
9.1 Implementation at Tommy Hilfiger Europe.....	68
9.2 Addition of new features	68
9.2.1 Collaborative annotating.....	68
9.2.2 Drawing input per annotation.....	68

9.2.2 Summarizing, grouping and filtering annotations.....	69
9.3 Professionalization.....	69
Appendices.....	70
Appendix A: Creative Technology Design Methodology.....	70
Appendix B: Semi-structured user test	71
Appendix C: Ethical Reflection report.....	72
Immersive Augmented Reality and Artificial Intelligence: an ethical review.....	72
Appendix D: C# code.....	77
MainController.cs.....	77
AnnotationController.cs	78
HUDController.cs	84
ButtonInteraction.cs.....	87
SaveLoad.cs.....	88
PushPinController.cs	89
Appendix E: Concept designs for tutorial interface.....	90
References.....	95

Abstract

An augmented reality application allowing for annotation of 3D models suited for CAD, focused on the fashion industry, has been designed. The goal of the annotation application is to improve the performance of fashion designers. To achieve this, research has been done in the domains of augmented reality, interaction methods for immersive XR, cloth simulation and feedback methods. A state of the art research was conducted, where several interaction methods were evaluated based on selected criteria. Based on the state of the art research, several ideas were developed that are suited for the context of the project. These ideas have been evaluated and selected.

From the selection of the ideas, a specification has been made. The specification described to develop a prototype of an augmented reality application allowing for annotation, workout out in Unity using the Meta 2 AR device. The best interaction method that has been suited for the functionalities of the application has been determined after three prototype iterations. The interaction method that had proven to be best, was a self-developed 'pushpin' method, which makes use of the SLAM-tracking of the Meta 2 device.

Evaluation of the final prototype showed that the prototype is a successful design of an augmented reality application, allowing for reviewing via annotations of 3D garments designed via CAD software. However, before this product can be implemented in the correct context, several aspects have to be improved and professionalized.

Acknowledgements

This project would not have been possible without my supervisor and my critical observer, dr. J. Zwiers and respectively dr. R.J.F. Ordelman from the University of Twente. I would like to thank dr. J. Zwiers and dr. R.J.F. Ordelman for their support, feedback and helpful ideas during this graduation project. As my supervisor, Mr. Zwiers has been supportive and ambitious without losing perspective of the timespan of the project. Second, I would like to thank Mr. L. Kuipers, innovation manager at Hecla, the client of this project. The input of Mr. L. Kuipers has been extremely helpful and essential for a successful outcome of the project. Last, I would like to thank my family for giving me the opportunity to study what I study and to do what I love. It has been a great project and an educational time.

List of abbreviations

CAD	Computer Aided Design
AR	Augmented Reality
VR	Virtual Reality
XR	X Reality / Cross Reality
HUD	Heads-up Display
HWD	Head-worn Display
HMD	Head-mounted Display
PACT	People, Activities, Context and Technology
SLAM	Simultaneous Localization And Mapping
SDK	Software Development Kit
API	Application Programming Interface
UX	User Experience

1. Introduction

1.1 Context

“To design an annotation-interaction system, that supports the feedback processes of fashion design CAD software.”

The context of this project lies in the fashion design processes. We explore the possibilities of annotating cloth design using 3D Augmented Reality (AR) environments, with a goal of finding the best annotation model to be used in the context. The AR environment that the model has been annotated in exploits state-of-art AR hardware, in the form of Meta 2 glasses, that enable a wide field of view compared to existing solutions like the HoloLens developed by Microsoft.

In the fashion industry, design processes are relatively traditional. It is a long and expensive road from ‘mold’ to the finished product. This bachelor project focuses on the feedback-loop during the creation of a new garment. This research assumes that the design process makes use of state-of-the-art CAD-models [3] (3D-models) with proper textures for the best experience.

1.2 Problem statement

During the garment design processes in the fashion design industry, there is little room for feedback. For Tommy Hilfiger designers, there are two main internal events every year where garment designs are presented to the coworkers of the company. At these events, some informal comments about the new garments are mentioned, but since the garments are already published once and worked out in detail, no changes shall be made. Moreover, it is difficult for designers to give constructive feedback when not seeing what the end result of the garment could look like in 3D.

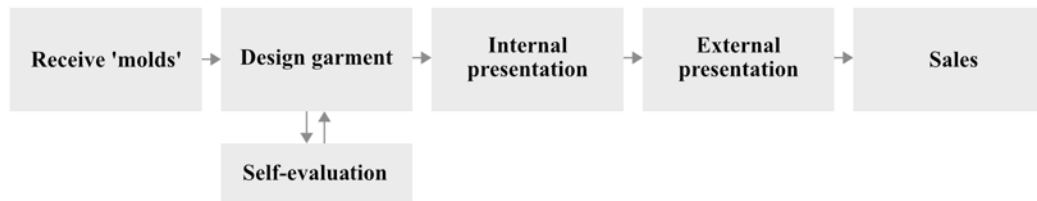


Figure 1.1 – Abstract structure of the design processes for Tommy Hilfiger Europe.

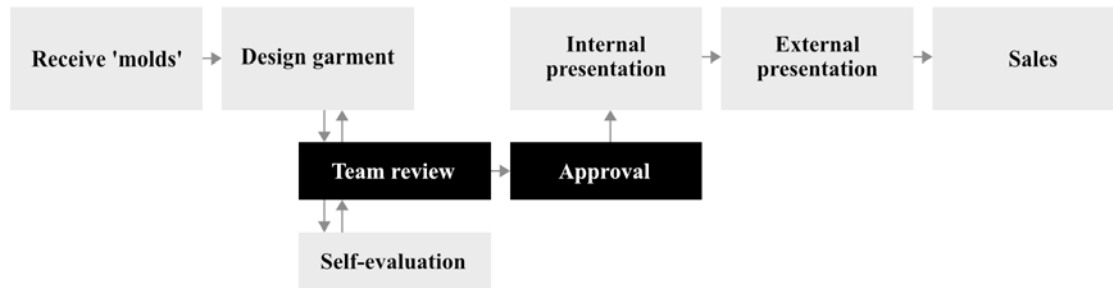


Figure 1.2 – Abstract structure of the design processes for Tommy Hilfiger Europe, after introducing the feedback application.

We propose to introduce a feedback loop supporting the design process, allowing to have a detailed look to the garment and to comment (annotate) on the garment, supported by novel AR techniques – “Augmented Reality (AR) is a technology which allows computer-generated virtual imagery to exactly overlay physical objects in real time’ [4]. The aim is to increase the design quality of the garments, decreases the one-time production costs for garments and speeds up the design process overall. The solution exists out of an annotation-interaction system, where a reviewer should be able to select a certain area of the garment design. After selecting an area of the design, the reviewer is able to provide feedback, hereby annotating an aspect of the garment. Feedback is given through voice input. Each annotation represents a point of feedback. On the other hand, the reviewee is able to see an overview of the given feedback and to select and view a specific point of feedback. The specific points of feedback allow for ‘replaying’ of the given feedback, by playing the voice input as it was recorded. The application prototype has been developed using the Unity SDK and has been applied for use with the Meta 2 Augmented Reality glasses. Reasoning for the made choices in this project are described further in this document.

1.2.1 Research questions

This problem statement has lead to the following research question:

“How to best design an interaction model to allow annotation of 3D-models in an augmented reality environment?”

For this research questions, several sub-questions are defined to properly structure the research process. These sub-questions are as follows:

“What methods of interaction are used by existing augmented reality installations?”

and

“Which interaction method(s) work best for annotation of 3D-models?”

and

“How can this/these interaction method(s) be applied in a way that is suitable for annotating 3D-models of garments to provide feedback for designers?”

2. State of the Art

Designing a multimodal interaction model to provide feedback during the design process of garments, requires some research in several fields. This research is done in the fields of virtual garment design, feedback methods, and tangible AR interaction. Lastly, an overview of the currently available hardware devices for AR development is given. Literature is stated to give a decent overview of what is currently researched and/or developed relating to this project.

2.1 Garment simulation

To be able to provide feedback on a garment design, the design has to be displayed properly. To simulate the prototype design of the garment in the AR-environment, high-quality and real time garment simulation is an important requirement. One of the first types of garment simulation date back to little over 30 years ago, such as Weil's approach in simulating an approximation of the proper shape and surface rendering [5]. The older approaches are relatively simplistic, as the performance of computing technologies was limited. Evidently, the better performance of computing nowadays (Moore's Law [6]) has led to more realistic, high-resolution simulations.

2.1.1 The early days

The first applications for computer-aided garment simulation started in the 1990's. A combination of several researched technologies such as cloth simulation, body modeling and animation [7], and collision detection and response [8]. These applications show virtual garment patterns, sewed together around a character.

2.1.2 Recent advances

Over the years, the accuracy and efficiency of the garment simulations were optimized. Mechanical models, reproducing the mechanical behavior of cloth were advancing in quality, where several implementation techniques originated. Researchers found that the most efficient and simplest way to properly display a garment with proper mechanical behavior was through particle systems [9]. However, such accurate models did take a lot of processing power as the required calculations are numerous. To be able to show the proper mechanical behavior in real-time, the efficiency of these models had to increase significantly. Desbrun et al. allowed fast simulation of mechanical properties of cloth. However, the computation speed still remains relatively slow for more complex garments, as the number of polygons are still limited due to the processing power.

2.2 Tangible interaction in Augmented Reality

For every AR-application, interaction designers are facing a new challenge: to create an AR interface that provides rich interactivity, in which the virtual space is emerged into the physical space as best as possible. Although the first AR-interfaces were developed nearly 40

years ago [2], interaction design for AR in three-dimensional environments is a relatively new and unknown field to designers. Earlier researches mostly are focused on the actual visual augmentations of the reality [3] via heads-up display systems (HUDs) [4], head-worn displays (HWDs), head-mounted displays (HMDs) or other wearable technologies [5]. Evidently, to be able to interact with the information projected via one's HUD, HWD or HMD, firstly the information has to be displayed properly.

Although in comparison to visual augmentation using HUDs, HWDs or HMDs, the number of studies regarding tangible interaction interfaces for AR-applications is relatively low, there are still numerous different interaction techniques researched throughout the years. These systems make use of various interaction techniques, ranging from eye tracking [6], finger tracking [7] or hand tracking [8] to voice input [9]. This literature review discusses the existing body of empirical research of interaction design in Augmented Reality applications, with the goal of determining which types of interactions are best suited for which purposes. To reach this goal, several peer-reviewed research papers describing ways to interact with AR interfaces will be discussed and their results will be relativized to the overall outcome of these papers.

Abstractly, all tangible interaction techniques can be divided into two categories: tracking-based and controller-based. Most of the tangible interaction techniques are developed for specific applications, meeting a specific requirement for the application. For example, the flexible pointer, researched by Alex Olwal and Steven Feiner, was presented to improve the selection and indication progress while manipulating 3D-objects [10]. Likewise, FingARTips is a gesture interaction technique researched by Buchmann et al. to improve the interaction with virtual content using hand gestures [8]. Many other specific techniques have been researched with the purpose of improving a certain aspect of existing interaction techniques or developing an entirely new technique.

All of the interaction techniques discussed in this literature review, can be divided into two categories: tracking-based and controller-based. Each of the mentioned interaction systems will be evaluated based on specified criteria: performance, applicability and ease of use. Later, the evaluation process for each criterion is explained in more detail.

Firstly, most interactions in AR-environments are tracking-based, making use of advanced tracking systems. These tracking systems exist out of sensors that can perform exact eye-tracking, finger tracking or hand tracking. In this paper, all of the techniques that primarily require precise tracking will fall under the category 'tracking-based interaction'.

Secondly, there are AR-systems that are controller-based, meaning they require a separate physical device. Other than the HUD, HWD or HMD, required for an immersive augmented reality experience, these systems make use of (a) separate controller(s) to help the user with providing input to the system. These controller-based systems fall under the second

category 'controller-based interactions' and will be discussed and compared to the tracking-based interaction systems.

To discuss the findings of every study discussed in this paper, every interaction system will be looked at critically. Criteria for evaluation of the systems are: performance, applicability and ease of use. Each criterion will be evaluated using a simple scale: low, medium and high. The first criterion, performance, refers to the precision and abilities of the tracking-based or controller-based system. The more precise a user is able to perform a 'select' action, the higher the precision is, thus the better the performance. Following, the 'applicability' criterion refers to the extent to which the system is applicable to every-day usage. For example, if the system requires lots of calibration – for every use – the system is not highly applicable. Lastly, the 'ease of use' criterion speaks for itself: the easier the system is to use, the better the ease of use is.

Interaction techniques in 3D-environments are all about positioning. For input by hand or fingers to work, the AR-system must know where the user's hands or fingers are in the environment. Furthermore, the position coordinates of the user's hands or fingers must be mapped properly to the digital environment, to be able to connect both worlds. In order to achieve a proper mapping of the coordinates, AR-systems make use of many sensors that are able to gain this necessary context-awareness. The more precise the positioning, the more accurate the interactions with the virtual world can be. The faster the tracking process takes place, the better the AR-experience will be.

Opening, several researchers that have designed tracking-based interactions in their studies will be discussed. Following, controller-based interactions will be discussed and evaluated.

First, Crowley, Berard and Coutaz [7] researched a fairly primitive form of Augmented Reality, referred to as the 'digital desk'. The digital desk set up consists of a video-projector, positioned directly above and facing the table, projecting a computer screen onto the table. This primitive form of AR comes with its drawbacks: a change in viewpoint of the user (projector, table) is merely limited to rotation. J. Crowley uses an algorithm that can detect the position of a device, finger or object in 2D-space. The device, finger or object must be first 'calibrated' into the system via a reference image. Unfortunately, this causes a drawback: every different object to track must be calibrated. The algorithm is heuristic, meaning that besides detecting the *current* position of the finger, it also calculates what the *next* most likely position of the finger will be. The most evident limitation of this tracking is the lack of ability to detect the object in 3D-space, which lacks the ability to be able to work with 3D-models or a 3D-environment. Actions such as 'mouse-down' were improvised in this research, by simply using the space bar of the keyboard of the computer that was running the system.

Evaluating the 'digital desk', there are several things to conclude. First, the calibration process required for every object, combined negatively influences the applicability, hence scoring a low for applicability. Second, the performance of the system is low, despite the heuristic function the tracking process appeared to be imprecise. Finally, the improvised 'mouse down' actions result in a low score for 'ease of use' as well. All in all, the 'digital desk' by Crowley, Berard and Coutaz did not perform well.

FingARTips takes fingertip-tracking just a bit further, by placing fiducial markers on the hand of the person, and a simple haptic feedback device to be able to 'feel' virtual objects and interact with them [8]. The researchers focus on gesture-based interaction, a technique that has been popular in VR environments, while being less common in AR applications. Buchmann et al. researched a system allowing grabbing, pressing, dragging and releasing interactions. They put the system to use in an urban planning workspace, where a single hand suffices for all the necessary interactions. Buchmann et al. created a glove, containing the simple haptic feedback device and having three fiducial markers attached to it. Other than the 'digital desk', the FingARTips system does not require calibration for each object to detect, as it merely has to be calibrated once to calibrate the glove. The markers exist out of black and white squares, similar to the more familiar QR-codes. Via the open-source ARToolKit library, the position of these markers and their relative distances are calculated. Using two algorithms, gesture coordination is handled. In the glove developed for the system, the buzzer provides haptic feedback when a virtual button is pressed by the user. The main drawback of this system is having to put on gloves with special markers for the system to work. The advantage of this tracking method is the advanced gesture coordination possibilities. A simple one-handed glove provides numerous interaction methods.

Evaluating, FingARTips has done a significantly better job than the 'digital desk'. The ability to track a hand in 3D-space, where 'digital desk' was only able to track objects in 2D-space, results in a medium score for performance of FingARTips. Secondly, the lack of calibration required for every startup of the application, results in a medium score for applicability. Unfortunately, the glove with the fiducial markers does have its drawbacks: having to put on a glove decreases the 'freedom of hand', therefore resulting in a low score for ease of use.

Falling under the second category, controller-based interactions, some applications require an advanced control over input, where precision in positioning is a high priority. A separate physical controller, can provide this precision. There are two categories that divide the physical controller interaction: dumb and smart devices. The devices falling under the 'dumb' category, contain almost no technology and merely act as a support for the system to work. The devices falling under the 'smart' category do exist out of technology such as sensors or analog buttons or controls.

The approach from T. Kawashima et al., for instance, consists out of a dumb controller, to be named 'Magic Paddle'. In their prototype, the controller exists out of a piece of carbon,

similar to the shape of a traffic controller. The Magic Paddle contains a black square, so that the system can track its position properly. Using the Magic Paddle, virtual objects can be moved from place to place. The Magic Paddle is an 'accessory' to a book, where several distinguishable squares printed on the book make the system render virtual objects. The Magic Paddle can then be used to move the objects around. This interaction approach is fairly limited, as it does not offer any way to particularly interact with the shown 3D-models, other than moving the objects.

Evaluating the Magic Paddle, the system has its perks and flaws. Unfortunately, there was little information concerning the performance of the Magic Paddle. Considering the abilities of the system, as it does not offer any way to interact with the 3D-models, the score for performance can be noted as low. Furthermore, the applicability is limited to books, it is straightforward to also criticize the applicability of the Magic Paddle as low as well. However, the Magic Paddle application does its job as an accessory for books, which results in a medium score for applicability, in its field. The ease of use score can be argued to be high, as the system is fairly limited in its interactions, the usability of the system is extremely high – as the actions to perform are easy to perform. Therefore, Magic Paddle does not outperform the systems mentioned earlier, however, it does prove to be extremely easy to use.

Continuing, the approach of Wright et al. take the usage of a physical controller a step further. Their research lies within the medical field, where already plentiful of augmented reality projects have been researched to provide surgical simulation. This particular project makes use of a Leap Motion (LM) hand controller, which is a device that detects the movement and position of the user's hand when held above the sensor [14]. Usually, the Leap Motion controller is used for virtual reality environments, as it was developed to work with these systems. However, the approach of Wright et al. combines the virtual reality controller with Vuforia, the aforementioned 'simplistic' augmented reality technology, to transform the LM controller to work with augmented reality. The controller is compared to a popular physical controller in the medical field named 'NeuroTouch'. The NeuroTouch controller does not work with augmented reality and is merely a physical controller. The LM hand controller has a fingertip position accuracy of 0.01 mm, making the controller extremely accurate [14]. Evidently, this accuracy is of high importance in the medical field. Wright et al. report a more intuitive 3D interactive experience using the LM controller, rather than the NeuroTouch stylus. Wright et al. also report that their affordable, easily accessible simulator has great potential for future use, when it is tested further and the tracking accuracy is improved.

Evaluating the approach of Wright et al., the system scores a definite high for performance, as the LM controller offers a precision of 0.01 mm, which is a requirement for the specific context of the application. Furthermore, the applicability can be argued to be high, even though the system is not applicable to every context, little calibration is required for every usage. Following, the ease of use can be argued to be medium, as the study showed a steep

learning curve, meaning that users were getting exponentially better and better over time [14].

Unfortunately, it is difficult to determine which of these interaction types is the most time-efficient and productive, as all of the mentioned applications have different requirements and different measures of success. Even so, not every research project has measured 'task completion times'. Nor had every project the possibility of any quantifiable measure. Therefore, for each discussed paper, the results will be discussed briefly to determine which of the used interaction methods is the most practical and intuitive.

Firstly, the digital desk by J. Crowley, F. berard and J. Coutaz in 1995 [7], has quite a large limitation. Since the tracking system is limited to detect an x -and y-position in 2D-space, there is no real interaction possible in 3D-space. Hence, this interaction method can be stated as unsuccessful, even though it was a success at the time. Furthermore, the three 'low' scores for the earlier-mentioned criteria do not help either.

The approach of Buchmann et al. has more potential, as it allows users to grab, drag, rotate, drop, push and point objects in 3D-space. Moreover, the researchers report a low learning rate. However, the usage of fiducial trackers does have a specific drawback: the requirement to wear a glove containing these trackers is impractical and the fiducial trackers seem to not always work properly, causing usability concerns [8]. The ability to have haptic feedback did provide good results, while the absence of depth cues made interaction difficult. Compared to the digital desk, the approach of Buchmann et al. is a step forward, though it lacks ease of use.

The 'Magic Paddle' dumb controller, researched by Kawashima et al., is a limited approach. The developed system also makes use of fiducial trackers, showing similar problems to Buchmann et al. The interaction required a piece of carbon containing a fiducial tracker without any sensors or other technology, making the controller device 'dumb' and its interaction methods fairly limited [15]. However, the Magic Paddle resulted in a low - medium - high score for performance - applicability and respectively ease of use, meaning that it is a large step forward regarding the 'ease of use' criteria, compared to the approach of Buchmann et al.

Seemingly, the approach of Wright et al. works best. A high - high - medium score for the criteria is the best so far. In their research, they state that the LM controller has a high accuracy of 0.01 mm, making its potential use in the medical field possible. Even so, the researches state potentially successful results, compared to the compared NeuroTouch device. The accurate fingertip tracking allows for plenty of interaction types [17]. The drawback is the need for the physical controller to be positioned somewhere, taking away a part of the 'immersive' augmented reality experience. Compared to the other approaches, the precision and applicability are massive steps forward.

All in all, there are several interaction methods discussed. The older methods (Magic Paddle [15], Table-top [16], Buchmann [8], Crowley, Berard and Coutaz in 1995 [7]) seem to have a preference for the use of fiducial markers, causing tracking errors and limiting the interaction possibilities. The newest interaction method, by Wright et al., using the Leap Motion controller device, shows best results and performance.

Coming back to the goal of this literature review, to determine which types of interactions are best suited for which purposes, several conclusions can be drawn. First of all, tracking-based interaction types can lack precision, reliability and ease of use, while excelling in applicability for specific scenarios. On the other hand, controller-based interaction types excel in precision and reliability, while they do cope with a learning curve – having a better ‘ease of use’ over time.

Which interaction type is best suited for which purpose, can be concluded. If the focus of the purpose is on precision: using a controller-based interaction will most likely be the most successful. Then again, if a desired goal of the application is more related to applicability, both tracking-based and controller-based interactions can be used. The applicability of an interaction system is highly dependent of the scenario of the application. Finally, if the focus of the application lies with ‘ease of use’, it depends on the frequency and duration the users will use the application. If users will use the application frequently and for longer periods of time, then a controller-based interaction system will provide best results. If, however, the users will only use the system once, for a short duration, tracking-based interaction systems are proven to be more intuitive and easy to use – hence providing better results.

2.3 Feedback methods

2.3.1 Feedback forms

Feedback can be given in different forms, such as written, verbal and numerical. Augmented reality adds another important level of feedback, similar to one-on-one conversations: non-verbal. ‘The format of feedback is often directly related to the context’ [26]. Written feedback related to a written assessment may be of greater help than verbal feedback. Even so, written feedback carries more weight rather than verbal comments. Feedback on paper should be short and brief, to not lose the message of the feedback. The solution to understanding long, complex feedback is scaffolding. By scaffolding, the user receives feedback along with prompts, clues, solutions and instructions. This divides the overall ‘improve’ task into several smaller tasks, to make it easier to understand and to act for the user [27].

2.3.2 Content of feedback

Proper feedback should contain the essence of what to improve, and how to do this. It should be reinforcing, as it guides the users on how to improve future performance. It is inevitable to neglect a person’s opinion in the feedback, as feedback is always based on opinions and suggestions.

2.3.3 Timing of feedback

‘Timing and frequency of feedback are equally important for quality of feedback to be delivered’ [26]. Feedback sessions are meant for improving the performance. Ideally, feedback related to a performance should be given as close to the event as possible. In the context of this project, this would conclude that the feedback should be given as soon as a new prototype is finished. The ‘learners’, or in context, the designers receiving feedback, should be actively involved in the feedback process.

2.3.4 Constructive feedback

There are several key features of constructive feedback. Firstly, the given feedback works best when the reviewer, is committed and engaged in the process. Therefore, to increase engagement and commitment to the prototyping process, the reviewer should be related to the same prototype over several frequent time periods. Once the reviewer starts noticing positive results related to the previous feedback session, the reviewer will be more committed to the garment prototype in particular.

Secondly, both the reviewer and the reviewee must be aware of the criteria that will be assessed. Even though in fashion design, most criteria given to the garments will be based upon opinions, there are still a lot of ‘standard’ criteria that should be met in a prototype. If the designer does not meet these criteria, this should be made clear by the feedback given by the reviewer.

Thirdly, the reviewer must focus on specific flaws or strong points, rather than giving general feedback. For example, feedback such as ‘It looks good.’ will do nothing to improve the performance of the designer, other than positively showing that the garment is approved, while feedback such as ‘The pattern on the chest pocket is well worked out and has a nice feel to it’ will make the designer recognize and memorize its positive performance.

2.4 Annotation in Augmented Reality

There have been found several applications and study’s related to 3D model annotations, while the number of study’s that were related to 3D model annotation in an augmented reality environment was limited. Alducin-Quintero et al. have analyzed the impact of 3D model annotations on CAD (Computer Aided Design) user productivity in the context of the New Product Development Process [28]. Their approach resulted in a 13-26% reduction of time needed to perform engineering changes in existing models [28]. The annotations in the research paper of Alducin-Quintero were intended to improve the understanding of the intentions of the design. The findings of Alducin-Quintero et al. show that the productivity impact through CAD annotations depends on several factors, such as the geometrical complexity of the 3D model, and the quality of information in the annotation.

The approach of Nuernberger et al. focuses on anchoring 2D gesture annotations, or in other words: enhancing original drawing gesture input to achieve visually-appealing drawn

annotation [29]. The researchers have acknowledged the problem of drawing in 3D, when the perspective of the camera is dynamic. Nuernberger et al. have presented a new approach to solving this problem. First, the system classifies which type of gesture has been drawn. Following, the gesture input is optimized in a way that conforms more to the original intention of the user. Unfortunately, the researchers have only focused on two types of gesture annotations: arrows and circles. Arrows are anchored via analyzation of 2D and 3D image input (single image: snapshot provides 2D information while images over time provide 3D information), while circles are anchored via a relatively simple algorithm.

2.5 Hardware

2.5.2 AR versus VR

Evidently, to be able to create a working prototype, a device to support the application is necessary. To determine which device is most suited to the context of the application, first the differences between AR and VR must be made clear. The most important advantages and disadvantages are shown in Table 2.1.

Aspect	AR		VR		Win?
	Advantages	Disadvantages	Advantages	Disadvantages	
Processing power	-	Requires 'a lot' of processing power	Requires less processing power than AR-devices	-	VR
Visuals	High-quality display and textures	Low comes with limitations in virtual environment size	FOV with in	Low-resolution displays	AR
Contextual awareness	Complete contextual awareness	Can distract the user from the application	Better ability to focus	No contextual awareness	AR
Interaction	Intuitive hand interaction: hands free	Specific tracking is difficult to calculate	Mostly controller based: specific	No hands-free interaction	AR

Table 2.1 – Advantages and disadvantages for AR versus VR devices per category.

As shown in table 2.1, the AR wins on three of the four categories. VR devices require less processing power, hence the VR group wins in this category. The advantages and disadvantages for each category are based on the top-of-the-line device for AR and respectively VR. The advantages of the high-quality visuals and interaction categories are based on the top-of-the-line AR device, which supposedly is the Meta 2. Contextual awareness for AR opens the door to many possibilities, where collaboration is the key advantage. Figure 2.1-a displays the lack of contextual awareness in VR environments best, compared to Figure 2.1-b, where the environment can be perceived as if the user is not wearing any device at all. From this brief research can be concluded that AR is most suited for the application.

2.5.3 Chosen hardware for prototype

Now that AR has been selected as the scope for our application, the following aspect to be researched is *which* AR device to use. In table 2.2 the most common AR devices that are currently available for sale are displayed.

Name	Developer	Advantage	Disadvantage
HoloLens	Microsoft	Extensive SDK, spatial sound, gesture input, voice support, wireless	Low FOV, heavy
Atheer AiR	Atheer	50-degree FOV, accelerometer, gyroscope, magnetometer, gesture interaction, wireless	Simplistic, 2D-UI Low FOV
Meta 2	Meta Co.	90-degree FOV, Hand interactions and positional tracking	Wired
SmartEyeGlass	Sony	Many sensors (accelerometer, gyroscope, compass, brightness sensor), microphone input, wireless	Low FOV
Smart Helmet	DAQRI	High FOV, infrared cameras, depth tracker	Smartphone usage

Table 2.2 – An overview of the most common augmented reality glasses.

The two devices that stand out in this table are the popular HoloLens and the Meta 2. The HoloLens stands out because of its popularity, where the Meta 2 stands out for its recent release date. Due to the popularity of the HoloLens, many libraries are available, a broad documentation allows for a quick understanding of the software kit and the large community can be of help during the programming. One can say the opposite for the Meta 2 device, as the device is new, the support compared to the Meta 1 and the HoloLens is less extensive.

However, it possibly will pay off to look into the Meta 2 further, as the specifications of the device are remarkable. The device offers a 90 degrees FOV (field-of-view), the largest of any of the listed devices. Even so, the high-quality display of 2560 by 1440 pixels allows for high definition visuals in the virtual overlay. Compared to the maximum FOV and display quality of the HoloLens, the Meta 2 stands out.

Figure 2.1-b and 2.1-c sketch an idea of the difference in FOV between the Meta 2 (2.1-b) and the HoloLens (2.1-c). As can be seen, the lower FOV results in a less immersive experience, limiting the possibilities of the virtual overlay. The high-quality display, the large FOV and

the Meta 2 SDK were the key factors to decide that the device that will be used for the prototype is the Meta 2.



Figure 2.1 (a, b, c from left to right) –

- a: Sketch of application in VR environment,
- b: Sketch of application in AR environment supporting the Meta 2 device,
- c: Sketch of application in AR environment supporting the HoloLens.

2.5.4 Background information of the Meta 2

The Meta 2 is the second version of the AR-glasses that have been developed by the team of Meta Co. The Meta Company started in 2012, as the founder of Meta Co. envisioned a world without screens. In 2013 the team of Meta launched a Kickstarter crowdfunding campaign, pledging \$194,444. Not long thereafter Meta launched its first development kit. The second version of the Meta glasses, Meta 2, was announced in February 2016. The Meta 2 glasses feature an industry-leading field of view of 90 degrees, a 2.5K screen resolution and direct hand interactions.

2.5 Conclusion

Several conclusions can be made from the state-of-the-art chapter for this project. First, garment simulation has been researched thoroughly and therefore a proper high-quality garment simulation should be feasible to implement in Unity 3D for the prototype. Garment simulation is a key subject, as it should replace the physical production-quality garment with a virtual garment. However, the focus of this project lies in the annotating of 3D-models, so the garment simulation can be put aside, to a certain extent.

Second, interaction methods for augmented reality have been researched extensively as well, over the past decade. A few methods such as Magic Paddle [23], table-top [24] or the approach of Buchmann [16] or Crowley, Berard and Coutaz [15] make use of fiducial markers, as the tracking solution for these markers are relatively easy to work with. Making use of an external controller, such as the Leap Motion controller, allows for great precision in the interaction input. It is still unclear what the precision is of the current tracking methods of the Meta 2 wearable device, this has to be researched in the project.

Third, providing feedback can be sorted out to the depth, criticizing the message of the feedback or the form it was given in. In this project, there has to be set a limit for the requirements of the feedback, to be able to focus on annotating. The feedback that will be given by the users of the prototype will most likely be verbal feedback, recorded with an

external headset. Henceforth, there should be researched what other forms of feedback would be more suitable, as the time limit of this project does not allow implementing this.

Moreover, Alducin-Quintero et al. have analyzed the impact of 3D model annotations on CAD (Computer Aided Design) user productivity in the context of the New Product Development Process [28]. The research shows that an increase in productivity is possible via decent annotation methods. Furthermore, as Nuernberger et al. have acknowledged: the problem of drawing in 3D, when the perspective of the camera is dynamic was difficult to solve. Nuernberger et al. have presented a new approach to solving this problem. Most likely, implementing this approach or a variant of this approach will be difficult and perhaps not feasible. Whether implications should be made in implementing annotations via drawing input, should be researched.

Concluding, all of the above-mentioned aspects require further research. The existing work has shown that it is possible to create a system that meets the requirements of this project. To be able to create a working prototype of a user interface allowing annotation of 3D models, every required step should be well thought-out and researched.

3. Methods and techniques

To be able to answer the research questions of this project, it is necessary to design different prototypes with different interactions, where each user interface has a slightly different UX design. For each prototype iteration, an experiment can be conducted via user testing. Evaluating these results in a general conclusion and a discussion of future work.

3.1 Design processes

The design processes that will be used in this graduation project, consists of four phases: Ideation, Specification, Realization and Evaluation. The phases are documented in the research paper by Mader and Eggink [30]. Creative Technology students of the University of Twente often make use of these methods (Design Methods of Creative Technology) during their graduation projects. When following these processes, a well-defined, well-researched prototype will be the result.

The Design Methods of Creative Technology makes use of a combination between a classical model for creative design processes: divergence followed by convergence and a more modern design approach: a spiral model. In the classical model, described by Jones in 1970, firstly the design space is opened up and defined, in which broad, out-of-the-box ideas are thought of and written down by the designer. Secondly, during the converging phase, the design space is 'reduced', until a specific solution or problem is reached. Between the divergent phase and the convergent phase, design decisions have been made based on the knowledge of the designer.

In the spiral model, however, the sequence of steps (divergence, decision and convergence) are interwoven, not allowing a logical order. 'Each design problem unfolds a sequence of questions that is specific to the starting problem and the context', according to Mader and Eggink [30]. In the Design Methods of Creative Technology, the interwovenness of design and knowledge questions is emphasized, meaning that outcomes of the last phase can have an influence on the phase before that, and so forth.

Firstly, the starting point of the Creative Technology method is the ideation phase. The problem statement is determined, followed by one or multiple brainstorm sessions where creative ideas come forward. Sources of inspiration for the creative ideas are related work and the outcome of the brainstorm session. The result of the ideation phase is an elaborated project idea, together with the problem requirements. Moreover, ideas on the interaction and overall experience of the product or prototype are also part of the result.

Secondly, the specification phase allows for exploring the design space. A brief evaluation and feedback loop are applied, in which the designer prototypes, evaluates, improves and prototypes continuously. After satisfied with the current prototype, the evaluation will lead to a functional specification.

Thirdly, the realization phase will focus on creating a prototype. The ideas and requirements from the ideation and respectively the specification phase have led to a clear implementation. For this project, the realization phase will focus on creating a user interface to work with the Meta 2 in Unity3D.

And lastly, after the prototype has been constructed, the evaluation phase allows for the testing of the prototype and reflection on the design processes. Two types of evaluation will be concerned: functional and user evaluation. The results of this evaluation will lead to a general discussion and a consideration of possible further research for the future.

3.2 Methods

3.2.1 Stakeholder analysis

A stakeholder identification and analysis is performed, with the goal to understand the environment and context of the project. The analysis will point out who is involved and possibly affected by the project. Moreover, it will identify the end-users of the project. The stakeholder analysis will be performed based on theory of Dix et al. [31] and Sharp [32], where the definition of a baseline stakeholder is cited as ‘anyone whose jobs will be altered, who supplies or gains information from it, or whose power or influence within the organization will increase or decrease’.

According to Sharp et al. [32], the most important group for the stakeholder identification is called the ‘baseline stakeholders’. The people involved in this group have the most influence over the product and vice versa. In the theory by Sharp et al. [32], the group of stakeholders exists out of four groups: users (distinguished by primary, secondary and possibly tertiary groups), developers, legislators and decision-makers. Evidently, the user group refers to the possible end-users of the product. The developer group refers to the technical experts that are involved: programmers and technicians. Furthermore, the people that fit in the ‘legislators’ group are those who could affect the product through rules and regulations. Lastly, the decision-makers group exists out of all related individuals or groups that have the largest influence within the management of the company that will be using the product.

3.2.2 PACT analysis

PACT stands for ‘People, Activities, Context and Technology’, acknowledged by Benyon et al. in 2005. A PACT analysis has been worked out during the ideation and respectively the specification phase. Performing a PACT analysis is useful to be able to better understand the current situation and to determine where there is room for improvement. Placing the product or prototype in a concrete situation allows for greater resemblance to the ‘real’ context. Theory of Moran et al. has been used to support this analysis [33].

3.2.3 Product and user requirements (MoSCoW)

In order to make the product idea more clear, the product and user requirements will be documented. The requirements are divided into two separate categories: functional and

non-functional requirements. Because of the limited timespan of this project, it will be unfeasible to implement all of the possible functionalities. Therefore, the list of requirements will be prioritized using the MoSCoW method developed by Dai Clegg [34]. The MoSCoW method is a prioritization technique used in management and often-used in software development. The MoSCoW term is an acronym derived from the first letters of each of the four prioritization categories: 'Must have', 'Should have', 'Could have' and 'Won't have'. At the end of this project, the final prototype will feature all the 'Should have' and 'Could have' requirements.

3.2.4 User and expert interviews

During several phases of the project, several user interviews have been conducted. These interviews differ in structure. Firstly, there have been the 'open' interviews for general feedback. Later in the process more structured expert interviews have been conducted. Regular feedback meetings with Hecla or PVH fall under the 'open' interviews. The results of these interviews have been discussed in the 'evaluation' section' and possible alterations to the prototypes have been mentioned.

4. Ideation

During the ideation phase, a concrete project idea has emerged from a brainstorm session and other inspiration sources. The project idea was meant to fulfill the needs set by the target group. In order to identify the needs set by the target group, the target group has been specified first. Via a stakeholder analysis, the requirements for the idea are made clear.

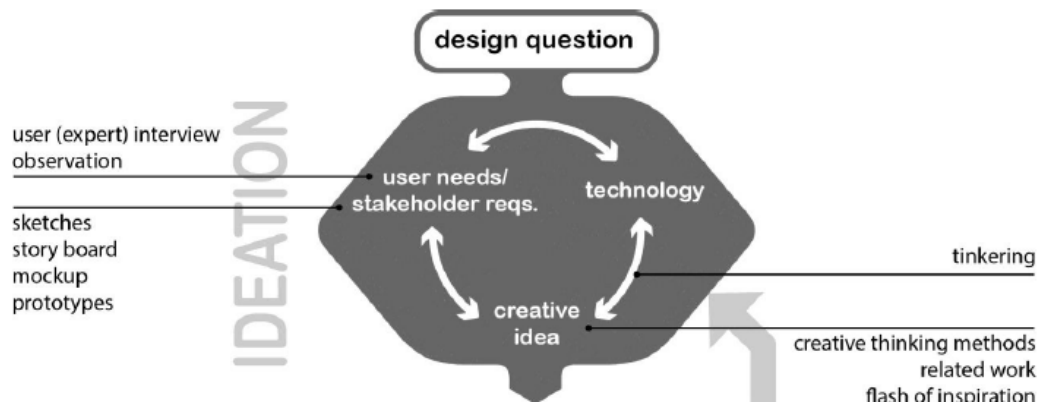


Figure 4.1 – The diagram for the ‘ideation’ phase found in ‘Design Methods for Creative Technology’ by Mader and Eggink [30].

4.1 Stakeholder analysis

To identify the stakeholders related to this project, the stakeholder identification method of Sharp, Finkelstein and Galal [32] is applied.

The stakeholders that have been identified are given in table 3. The stakeholders have been divided into four categories: users, developers, legislators and the decision-makers. A brief explanation of the companies is followed below. As shown in the first category, the product potentially has three types of end-users: designers and team leaders. Evidently, designers are the primary target group for this project, as the application is intended to be used by fashion designers. It is expected that in a later stage of development, the team leaders of the design groups of PVH will make use of the application as well, to further improve the performance of the fashion designers.

Besides the user category, other stakeholders fall in the developer, legislator and decision-maker categories. All of these activities will not fall in the timespan of this bachelor project, however. Since this project has the main goal of creating a prototype, all of the categories below ‘users’ are expected to be possibly relevant in the future.

	Role	Company
Users	Designers (reviewer and reviewee) (primary)	PVH
	Team leaders (secondary)	PVH
Developers	Programmers	Hecla
	Testing and maintenance	Hecla
	Installation	Hecla
Legislators	Quality Assurance	PVH
Decision-makers	Executives / management	PVH
	Designers	PVH

Table 4.1 – Baseline stakeholders, divided into categories

4.1.1 Hecla Professional Audio & Video Systems

The assignment for this bachelor project was provided by Hecla [35], a company situated in (amongst other locations) Hengelo, Overijssel. Hecla specializes in professional audio-visual systems and acts in the market of ‘audio-visual integrators’, where the goal of the company’s projects is to combine different audio-visual technologies to achieve the best possible solution for its clients. Their clients include some big names such as the *University of Twente* in the education field and PVH in the fashion industry.

4.1.2 PVH

PVH is a client of Hecla. The focus of this thesis was shifted, during the exploration phase, from Hecla to a combination of Hecla and PVH. PVH Corp (formerly known as the Philips-Van Heusen Corporation) [1] is a clothing company which owns brands such as Van Heusen, Tommy Hilfiger, Calvin Klein and more. The company originated in America over 135 years ago and is currently one of the largest global apparel companies.

4.2 PACT analysis

Use case 1: First usage of application

Title	First usage of application
Description	The user has received a 3D-model to review and wants to review the model.
Primary actor	Fashion design reviewer
Pre-conditions	User wants to review the model.
Post-conditions	User knows how to use the application, and is satisfied with reviewing the model successfully.
PACT	
People	Brad, a professional fashion designer of age 32 working at Fashion Company A is looking to review the garment design of a younger, newer colleague of the company.
Activities	Providing feedback via the review application using the Meta 2.
Context	The Meta 2 is attached to a computer, located in the same office where several designers work together.
Technology	Meta 2 AR glasses are used to display and interact with the 3D-model. Voice input is made possible via a gaming headset.
Scenario	<ol style="list-style-type: none">1. Brad enters the office, sees on his computer that there is a design to be reviewed, and sends the design to the Meta computer.2. Brad walks to the Meta 2 device, powers it on and puts the device on his head, such that the vision is clear.3. After putting on the Meta 2, Brad puts on the gaming headset.4. The applications boots up and the 3D-model to be reviewed is shown.5. The user selects the feedback button to start the review process.6. Once selected, the system starts recording.7. The user provides feedback.8. The user stops the recording.9. The user sends the review, stops the application and unplugs all of the devices.

Use case 2: Using the application to review a very specific area of the 3D-model.

Title	Reviewing a specific area of a 3D-model
Description	The user has received a 3D-model to review and wants to review a specific area of the model.
Primary actor	Fashion design reviewer
Pre-conditions	User wants to review a specific area of the model.
Post-conditions	User knows how to use the application, and is satisfied with reviewing the model successfully, having provided sufficient feedback for the specific area.
PACT People	Brad, a professional fashion designer of age 32 working at Fashion Company A is looking to review the garment design, in particular the chest pocket of the garment. The garment was designed by a younger, newer colleague of the company.
Activities	Providing specific and accurate feedback via the review application using the Meta 2.
Context	The Meta 2 is attached to a computer, located in the same office where several designers work together.
Technology	Meta 2 AR glasses are used to display and interact with the 3D-model. Voice input is made possible via a gaming headset.
Scenario	<ol style="list-style-type: none">1. Brad enters the office, sees on his computer that there is a design to be reviewed, and sends the design to the Meta computer.2. Brad walks to the Meta 2 device, powers it on and puts the device on his head, such that the vision is clear.3. After putting on the Meta 2, Brad puts on the gaming headset.4. The applications boots up and the 3D-model to be reviewed is shown.5. The user selects a specific area of the shown 3D-model.6. Once selected, the system starts recording.7. The user provides feedback.8. The user stops the recording and proceeds to review the 3D-model.9. The user sends the review, stops the application and unplugs all of the devices.

Use case 3: Viewing the given feedback of a 3D-model

Title	Viewing the given feedback of a 3D-model
Description	The user has received a reviewed 3D-model and wants to see and hear the given feedback.
Primary actor	Fashion designer
Pre-conditions	User wants to view the given feedback of the model.
Post-conditions	User knows how to use the application, and is satisfied with understanding the given feedback.
PACT	
People	Leroy, a professional fashion designer of age 24 working at Fashion Company A has made a garment design. His colleague, Brad, has reviewed the design. Leroy is curious to the given feedback and wants to see and hear it.
Activities	Selecting the given feedback per area, and listening and seeing the given feedback.
Context	The Meta 2 is attached to a computer, located in the same office where several designers work together.
Technology	Meta 2 AR glasses are used to display and interact with the 3D-model. Voice input is made possible via a gaming headset.
Scenario	<ol style="list-style-type: none">1. Brad enters the office, sees on his computer that there is a design to be reviewed, and sends the design to the Meta computer.2. Brad walks to the Meta 2 device, powers it on and puts the device on his head, such that the vision is clear.3. After putting on the Meta 2, Brad puts on the gaming headset.4. The applications boots up and the 3D-model to be reviewed is shown.5. The user selects a specific area of the shown 3D-model.6. Once selected, the system starts recording.7. The user provides feedback.8. The user stops the recording and proceeds to review the 3D-model.9. The user sends the review, stops the application and unplugs all of the devices.

4.3 Brainstorm

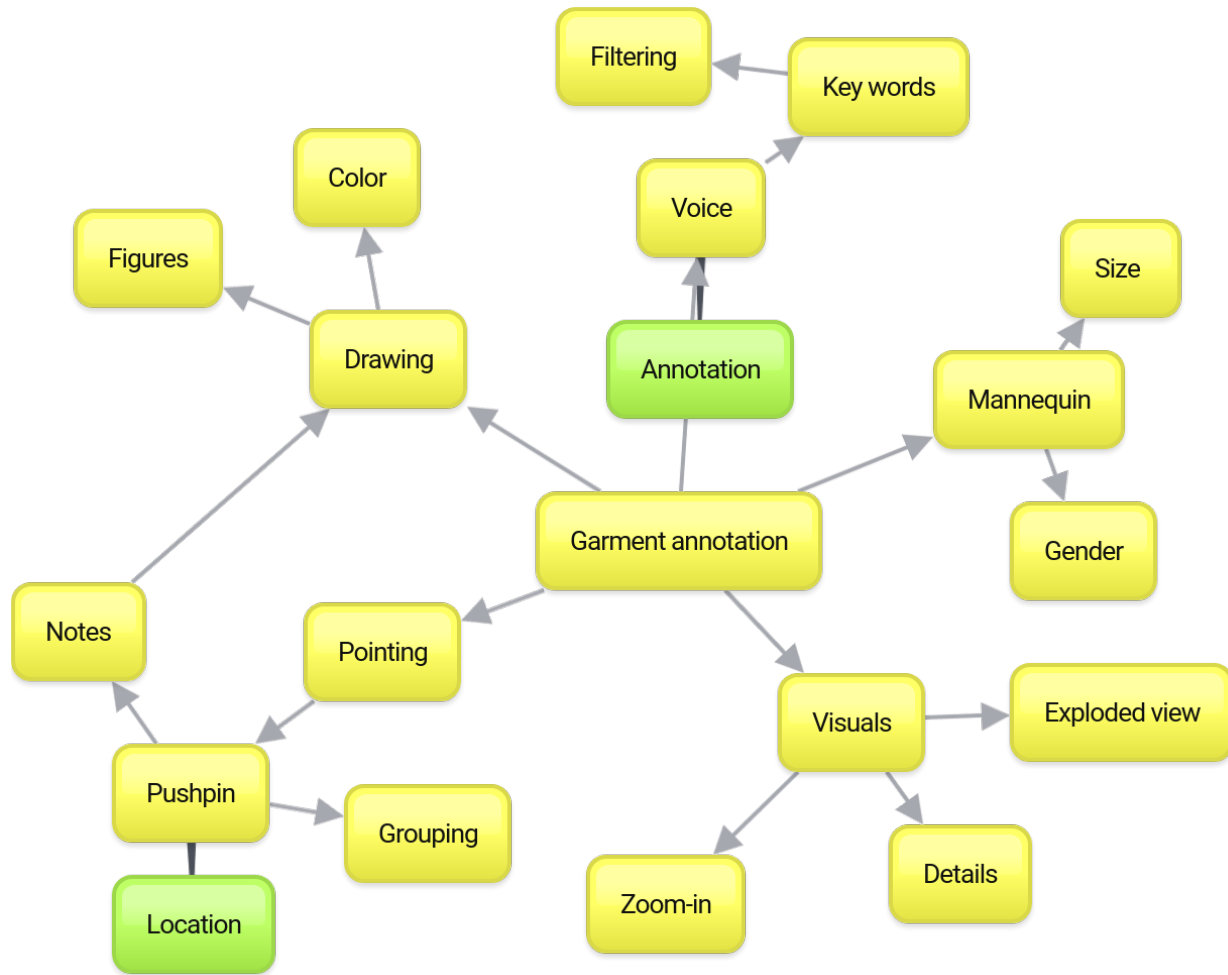


Figure 4.2 – Brainstorm diagram workout in an early brainstorm session.

4.4 User Interfaces

One of the most important factors of the design process for this application is the user interface. As the user is constantly interacting with the UI, it is a key factor. During the ideation phase, the requirements are not clear yet, therefore the sketched functionality of the user interface ideas differ a lot. Although the functionality requirements are not clear, several standard usability requirements are taken into account. Several wireframes were designed to show the general concepts of the user interface. The wireframes are made in 2D, where they will be displayed as a simple 'screen' with buttons in the AR environment, similar to the Meta 2 introduction interface.

4.5 Concept ideas

During the ideation phase, many concept ideas are thought out to 'spur' the creativity for the project. Most of these ideas are 'out-of-the-box', and infeasible for the current timespan of the project, hence these ideas are not worked out or added to the prototype. The future work section in the 'Discussion' chapter will continue on these ideas. Some ideas, however, are well-suited for the current context and are, therefore, added to the prototype.

4.5.1 Summarizing voice input in a word cloud

After the system has recorded a feedback 'moment', which the user has entered via voice input, the voice recording has to be played back in some way. Obviously, this can be done via direct playback of the recording. However, direct playback has certain disadvantages. First, the duration of understanding the content is as long as the recording, making it not time-efficient at all. Second, there can be several problems in understanding the voice recording, since the user can no longer interrupt the reviewer as if it were a real conversation. Third, if the user wants to hear feedback about a specific aspect of the garment (other than the area), the user has to listen to the complete feedback entirely.

To overcome these problems or disadvantages, the voice input can be analyzed and processed via an algorithm. There are several possibilities for transforming voice input into something understandable. One of the key ideas for this process was to 'summarize' the voice input, by targeting several keywords and displaying these, in the form of a word cloud, below the recording. Each keyword would represent a part of the recording or a combination of the recording. For example, the reviewer has mentioned the color of the chest pocket of a garment several times. The user would like to hear the feedback about the color in particular. To do this, the user can simply select the chest pocket area, where after he or she can select the 'color' keyword shown in the word cloud. The system then replays the specific parts of the recording containing the word 'color'.

4.5.2 Skype sessions during feedback recording

One of the key problems in the feedback loop over distance is not being able to have a one-to-one conversation. One-to-one conversations can be useful since the reviewer and the designer can tell each other their opinions, hereby resolving possible conflicts as fast as possible. To be able to have these one-to-one conversations over distance, an application such as Skype or similar to Skype can be implemented into the system. During the process of giving feedback, when the reviewer would like to know more about the decisions that were made, he or she can call the designer via Skype.

4.5.3 Different models for fitting

An advantage of the design application in AR could be to better see the fitting of the garment design for different postures or models. A 'Sims-like' selection procedure for the models would allow designers to design their garment to fit all postures. One of the difficulties of

implementing this feature is having to alter the 3D model of the garment so that it fits the chosen model, realistically.

4.5.4 Grouping annotations based on location

Having plentiful annotations for a single garment can make the interface unclear and unorganized. Therefore, an improvement to organize the interface further would be to group several annotations that are close to each other. These grouped annotations would show a slightly larger pin, possibly with the number of annotations attached to the pin.

4.5.5 Multi-user annotating

One of the advantages of AR over VR is being able to see the environment while working with the AR goggles. Hence, the AR technology allows continuation of the usual person-to-person interactions simultaneously with annotating the garment via the AR interface. Plenty of possible co-op interactions would be made possible, having multiple Meta 2 devices linked to the same software. User A could start annotating a certain aspect of the garment while user B supplements or criticizes this annotation.

4.5.6 Filtering of annotations

This idea continues on the first-mentioned idea: summarizing voice input in a word cloud. Once the voice input has been analyzed via speech-to-text software, every annotation can be tagged. A helpful addition to the application would be to implement real time filtering options, with which the user can 'highlight' or leave out a specific selection of the annotations. This allows the user to find the annotations related to the subject that the user is interested in more quickly.

4.5.7 'Step in' interaction with the garment

Augmented Reality has the main advantage of the user being able to see his or her environment. Therefore, the body of the user can also be seen by the user. To be able to see how the garment would fit the user, hypothetically speaking, the user could simply 'step into' the garment. A virtual mirror would then allow the user to see what the garment looks like on his or her personal body. The key difficulty for this idea is the virtual mirror: the software would have to combine the camera data with the virtual overlay with precise positioning. Moreover, the disadvantage would be for the user to see the limited quality of the camera in the virtual mirror, hereby almost fully neglecting the advantage of being able to see the garment as realistically as possible.

4.5.8 'Pushpin' selection interaction

An important aspect of this project is the selection procedure. During the review process, the user should be able to pinpoint the exact location of the annotation. The more precise, the better, as the location of the annotation can play a huge role.

4.5.9 Exploded view

Another visual-related idea is to implement an ‘exploded view’. This view method is often used in the mechanical industry. The exploded view can possibly have the similar advantages for the fashion industry. Allowing to let the user see the garment from the inside-out might be a way to perceive the details of the garment better.

4.6 Observations

4.6.1 General observations

Below are stated the general observations made during this project.

4.6.1.1 Drawbacks of Augmented Reality

Immersive augmented reality has many perks. The user can perceive an altered version of the real environment. Combining the digital world with the real world can simultaneously ‘improve’ the context, while maintaining the possibilities of the real world, such as one-to-one conversations. This is the main advantage of augmented reality over virtual reality, as the more immersive experience of virtual reality leads to a lack of engulfment of the real environment. However, there are several drawbacks for both virtual reality and augmented reality environments. One of the most important drawbacks is motion sickness. Having a perceived motion in the digital world, that is different than the actual motion in the real world, leads to a misunderstanding of context in the human brain, causing motion sickness.

4.6.1.2 Lack of reliability of the Meta 2 software

Unfortunately, the Meta 2 device has some disadvantages. As Meta Co. is relatively new to the market, the Meta 2 SDK documentation is fairly limited. Moreover, the Meta 2 device seemed to only work with the latest hardware. Some driver-related issues caused instable effects of the software, such as freezing every AR application after about one minute of uptime. Other than that, the Meta 2 software seems to ‘stutter’ on lower-end computers. When this stuttering behavior is experienced, it is simply not possible for the user to work with the AR applications as wished.

4.6.1.3 Tracking performance related to environment

The tracking performance of the Meta 2 device seems to be heavily related to the environment in which the device is used. A few requirements or demands are stated on the website of the Meta Co. which are required for the optimal experience with the Meta 2 device. Practical experience with the Meta 2 resulted in a few ‘ground rules’ for an optimal experience. These ground rules are a combination of observations and requirements from the Meta 2 website. The information is stated in table 3.

4.6.2 Interaction observations

The main focus of this application lies within the interaction with the user interface in the 3D space that is created by Augmented Reality. Since Augmented Reality is a high-tech solution, many necessities have to be met in order to make the technology work properly. The requirements for setting up the Meta 2 device to work properly, have been mentioned in Table 4.2.

Rule type	Description
Environment	The walls of the room should contain distinguishable, non-repetitive patterns.
	The room should contain a few objects with characteristic shapes.
	The room should have a size of at least 8 m ² for the user to be able to walk around slightly.
Lighting	The lighting should be similar to the lights of a 300 lum lightbulb. More lighting results in better tracking, yet slightly less visible AR objects. Less lighting results in slightly worse tracking and better visible AR objects.
Calibration	The Meta 2 device should be properly calibrated for every user. Even though the calibration process takes some time, this time-investment is worth every penny during application usage.

Table 4.2 – Observed requirements for setting up the Meta 2 device to work properly.

5. Specification

During the specification phase, the product design space is explored. The product requirements will be described based on prototyping and discussion with experts or end-users.

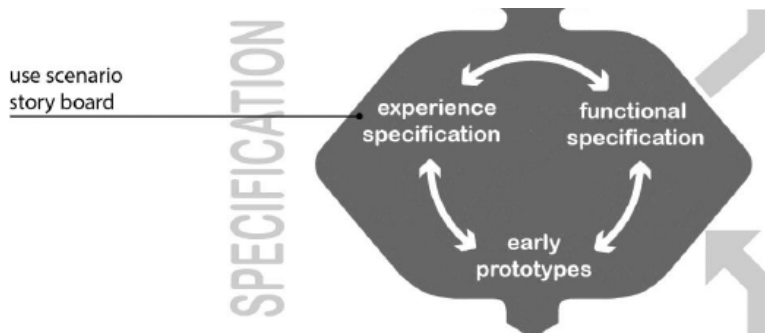


Figure 5.1 – The diagram for the ‘specification’ phase found in ‘Design Methods for Creative Technology’ by Mader and Eggink [30].

5.1 Prototype description

The prototype that has been built is a ‘3D-model-viewer’ application, allowing for annotation, developed with Unity using the Meta 2 SDK. The goal of the prototype is to be able to record several voice annotations, for specific areas of a 3D model of a garment. The application is written in C# and integrated to work with the Meta 2 augmented reality glasses. The interface of the application has been optimized for immersive augmented reality, mixing 3D interface elements with 2D interface elements.

Upon the first startup of the application, a basic tutorial, to be always-in-view, displayed via ‘flat’ imaging, explains how to use the application properly. The 3D model of the garment is displayed above and *anchored* to a round plate, which interacts as a *turntable* to be able to move or rotate the 3D model. The user is able to select a specific area of the 3D model via a pushpin selection interaction method, where dragging and dropping of a ‘floating’ pushpin allows for accurate selection of a location on the garment. After selecting the location, the user is able to start a new annotation, via a sequence of 2D interface elements that are located next to the location of the new annotation. The sequence of the interface elements existed out of the following actions: confirm location, start recording, stop recording.

An annotation exists out of a *location*, *instanceID* and a generated *id* of a pushpin on the 3D model, combined with a voice recording. The voice recording is linked to the annotation via the same *instanceID*. The voice recording is managed via an external headset with a high-quality microphone. Since the headset will cover the ears of the user, the audio output of the application should be mapped through the headset as well.

After there have been made several annotations, these annotations are displayed on the 3D model. Each annotation is represented by a pushpin on the related location. The user is able

to select a previously recorded annotation, by hovering his or her hand over the location to re-play the annotation.

5.2 Product and user requirements (MoSCoW)

From the outcomes of the feedback moments and the interview, a list of requirements can be constructed. The list has been separated into functional and non-functional requirements. Due to the timespan of the project, the focus has been set on the functional aspects of the prototype. As mentioned before, the prioritization of the requirements has been done via the MoSCoW methodology. The prioritized categories were (ranging from high to low priority): 'Must have's', 'Should have's', 'Could have's' and finally 'Won't have's'.

In the final iteration of the prototype, all requirements that have been prioritized within the 'Must have's' category, must have been implemented. All 'Should have's' were expected to be in the final prototype as well, if feasible. The 'Could have's' and 'Won't have's' are discussed later in this report, in the 'Future work' section.

No.	Requirement description	Priority
Functional requirements		
1	The user is able to wear the Meta 2 device and walk around properly in an area of at least 4 m ² .	Must
2	The user is able to read the instructions of the application	Must
3	The user is able to view the chosen garment through the application, in sufficient detail	Must
4	The user is able to rotate the chosen garment through simple turn-table interaction	Must
5	The user is able to walk closer and farther away from the garment, while the garment and table remain in the same place	Must
6	The user is able to select a specific area of the garment	Must
7	The user is able to start a voice recording after having selected a specific area	Must
8	The user is able to control the recording (play, pause, stop) when wanted	Must
9	The user is able to open and view the newly added annotation on the 3D-model	Must
10	The user is able to stop the playing recording of the annotation and go back to the overview	Must
11	The application contains 'grouping' functionality, where a relatively dense area of the garment with plenty of annotations is grouped into one larger pushpin	Should
12	The application uses speech-to-text software to summarize the annotations	Could
13	The user is able to filter the annotations based on subject or date	Could
14	The application allows to dynamically load a new 3D-garment with a separate set of annotation	Could
Non-functional requirements		
A	The user interface and interaction of the application should be clear and intuitive	Must
B	After the basic tutorial, the user should understand all the explained interactions properly	Must
C	The application should detect the users hands properly	Must
D	The user should be able to successfully create annotations	Must
E	The annotations should have clear sound recordings, with understandable speech	Should

Table 5.1 – Functional and non-functional requirements for the final prototype.

6. Realization

In this section, the technical and practical details of all iterations of the prototype are explained. After each iteration, a short evaluation shows the improvements to be made for the next iteration.

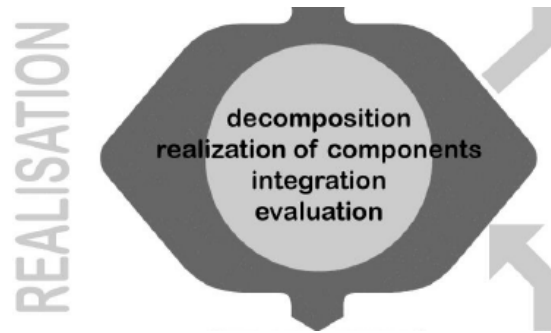


Figure 6.1 – The diagram for the ‘realization’ phase found in ‘Design Methods for Creative Technology’ by Mader and Eggink [30].

6.1 Prototype

There are several iterations that this project has went through.

6.1.1 First iteration

The first version of the prototype of the application started with simply testing out the hardware and software of the Meta. Since the software and hardware of the Meta 2 is relatively new and the documentation for the Unity SDK by the Meta Co. was fairly limited, the progress for the first prototype for the application was slow. The goal of the prototype was to display the garment properly and to be able to move the garment around and, if feasible, select a certain area of the garment to start annotation.

6.1.2 Second iteration

In the second iteration, a basic tutorial explaining the interaction of the application has been implemented, a selection method for annotation has been implemented and all requirements of the first iteration have been met. Furthermore, the second iteration has improved the issues found in the first iteration.

6.1.3 Final iteration

The final iteration of the prototype was meant to contain all ‘Must haves’ and ‘Should haves’. In the end of this project, a reflection has been made on the progress of the prototypes. The user was meant to be able to view existing annotations on the garment via pushpins. In order to view and listen to a recorded annotation, the user can select the pushpin, where an interface menu pops up. The interface menu allows the user to start pause or stop the recording.

6.2 First iteration

The goals of the first iteration are to set-up the Meta 2 device to work properly with Windows 10 and Unity, start programming the application and to implement a 3D-model of a garment designed by PVH. All requirements for this prototype, set up by the researcher, are stated in table 6.1. Following the table, a paragraph explaining the steps involved during the realization of this prototype.

6.2.1 Iteration requirements

No.	Description
1	Set up the Meta 2 device to work properly with Windows 10 and Unity
2	Create an application in Unity that allows to see objects through the Meta 2
3	Implement a 3D-model of a garment designed by PVH so that all visuals are similar to a real-life version of the garment
4	Implement an interaction to be able to move, rotate or resize the 3D garment

Table 6.1 – Requirements for the first iteration of the prototype.

6.2.3 Procedure

In order to set up the Meta 2 device to work properly with Windows 10 and Unity, the Meta 2 SDK should be installed on the system. The Meta 2 SDK installs all necessary drivers for the Meta 2, the Meta 2 Utility and extracts the Meta 2 assets required for Unity to work with the device. The Meta 2 Utility is software existing out of multiple programs. The software set contains a calibration program, a program to set up the display of the Meta 2 properly, a program to view the camera output of the Meta 2 (combined with the virtual overlay) and multiple tutorials and demos to start with.

After the successful set up of the SDK, and managing to get a grip and feel of the Meta 2 device by following the set-up tutorials and demos by the Meta Co, all preparations were made to start working on the first version of the AR application. After creating a new project in Unity and importing all assets of the Meta 2 SDK, the integration with the Meta 2 device can be established. Importing the *MetaCameraRig* into the project allows for integration with Unity and the Meta 2 device.

An important part of this iteration is to display a garment in the AR environment as realistic as possible. Due to the large resolution of the Meta 2 (2560 x 1440), the visuals of the ‘holograms’, as the Meta Co. names their AR overlay, are well-detailed. The problem for this step seemed to be to import a 3D model of a garment with the correct textures and mapping. Firstly, finding a realistic 3D model of a garment was more difficult than expected, as there were no free assets in the Unity Asset store or any other free 3D model assets website. After long communication with PVH, a 3D model of a garment was acquired in .obj format. Unfortunately, the importing of the .obj format was not as successful as hoped for, as the texture mapping was lost in the process. After troubleshooting, importing the .obj into

3DSMAX seemed to be the most successful. The .max scene created by 3DSMAX could be imported into the Unity project and a successful import of the 3D model of the garment was achieved, as seen in figure 6.2. The visuals of the 3D garment were astonishing and more realistic than expected, found by the researcher and user testing.

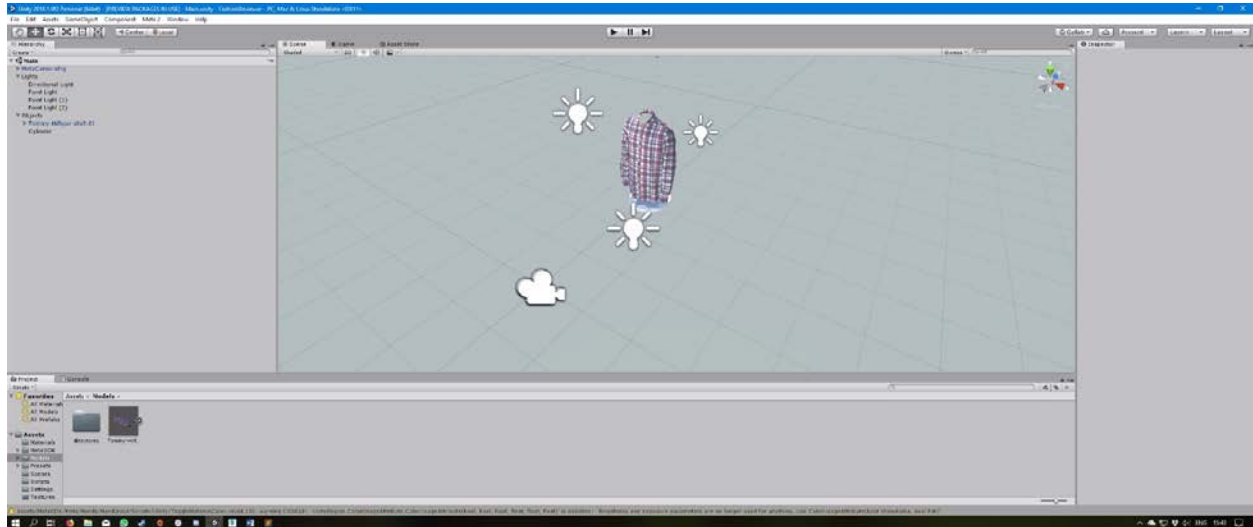


Figure 6.2 – Unity project set up to display the 3D model of the garment in .max format

After having successfully imported the 3D garment, the first three requirements have been met. The fourth requirement states that the user should be able to move, rotate or resize the 3D garment in the AR environment. There are several ways to do so. The first implementation to try out was to place a *Grab Interaction* script, which uses the Meta 2 SDK software to move, resize and rotate objects. The implementation requires a *Collider* trigger of the 3D garment, to recognize when the user is touching the garment. After testing out this implementation, a few issues became clear. Firstly, the *Collider* has to be really sophisticated for the *Grab Interaction* to work properly. Secondly, the hand tracking of the Meta causes unwanted interactions – rotating or rescaling the garment when not actually wanting to touch the garment. Thirdly, the *Grab Interaction* seems to make the 3D object ‘stutter’ when rescaling or moving the object. And lastly, it can be difficult for the user to rotate the 3D object via the *Grab Interaction* system, as it requires two hands, and sophisticated hand tracking, to rotate the object successfully. All in all, the *Grab Interaction* for manipulating the objects position, rotation and scale, can be concluded to not work intuitively and properly.

Therefore, another implementation had to be thought of. One of the ideas was to work with a turn-table, as the Meta 2 SDK also contains a turn-table interaction script, this seemed to be a clever solution. The turn-table, or ‘table-top’ or ‘round platform’ would be positioned below the garment, so that the garment is floating above the platform. Technically, rotation of the garment would be ‘locked’ to the rotational axis of the turn table, so that when rotating the turn table, the garment rotates at the same direction and speed. This idea was implemented and tested.

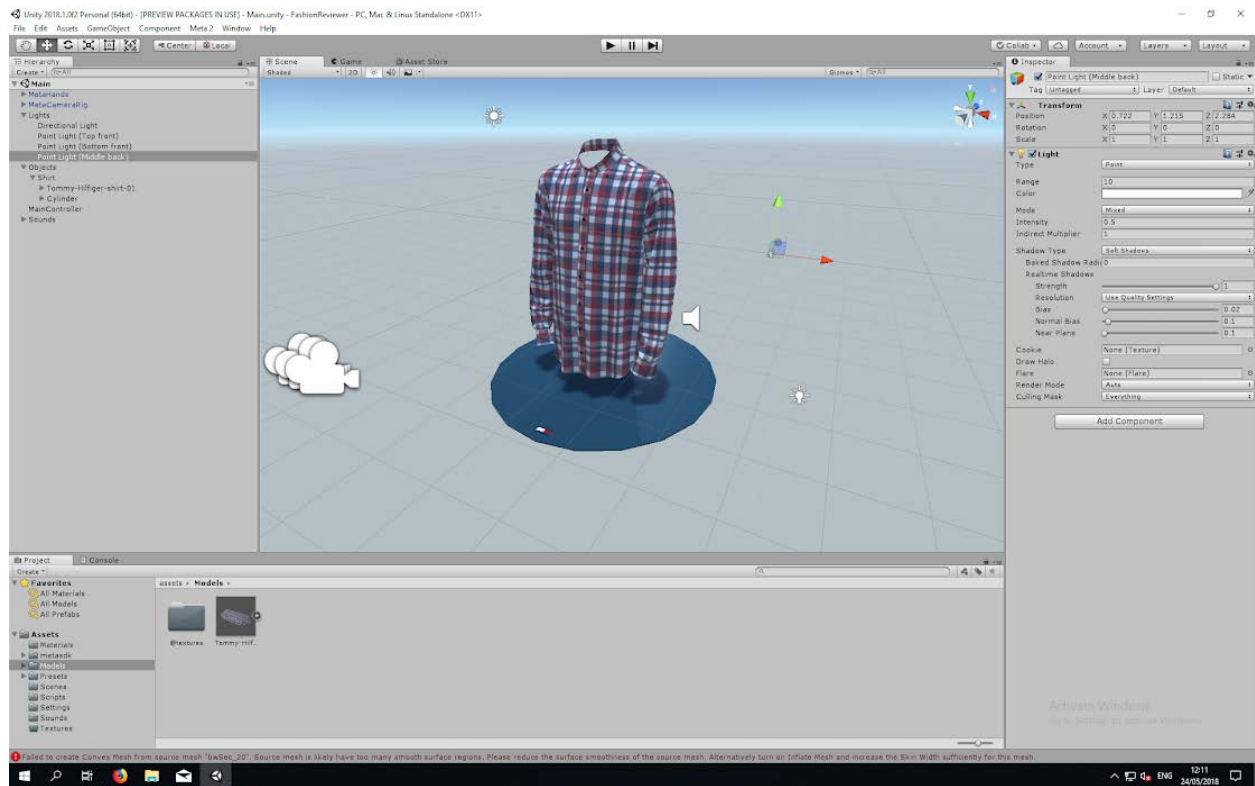


Figure 6.3 – The Unity project containing the table-top platform and the 3D garment

6.3 Second iteration

The goals of the second iteration is to be able to select an area of the garment for annotation. The selection procedure is critical, as the more specific a selection point can be, the more detailed the reviewer can provide its message. Furthermore, the second iteration will improve on the issues caused by the first iteration.

6.3.1 Iteration requirements

No.	Description
1	Improve the turn table interaction to exclude unexpected behavior
2	Implement a selection interaction in which the user can select a specific area of the garment
3	Implement a basic tutorial in the application to explain to the user how the interaction and the UI works

Table 6.3 – Requirements for the second iteration of the prototype.

6.3.2 Procedure

The first goal or requirement for this iteration is to improve the turn table interaction. The observed unexpected behavior was the system detecting trigger events for the turn table, when the hands of the user were not touching the table, combined with the system *not* detecting trigger events when the user *was* touching the table. After some testing, it was clear that these errors were mostly caused by the tracking system of the Meta 2. The hand tracking system of the Meta only ‘triggers’ an event when the hand is about 3 ‘virtual’ centimeters into the object. In an ideal situation, the action would be triggered when the user is *exactly* touching the object. To compensate for this behavior of faulty tracking, the most simply and effective solution is to increase the width and height of the *Collider* attached to the turn table.

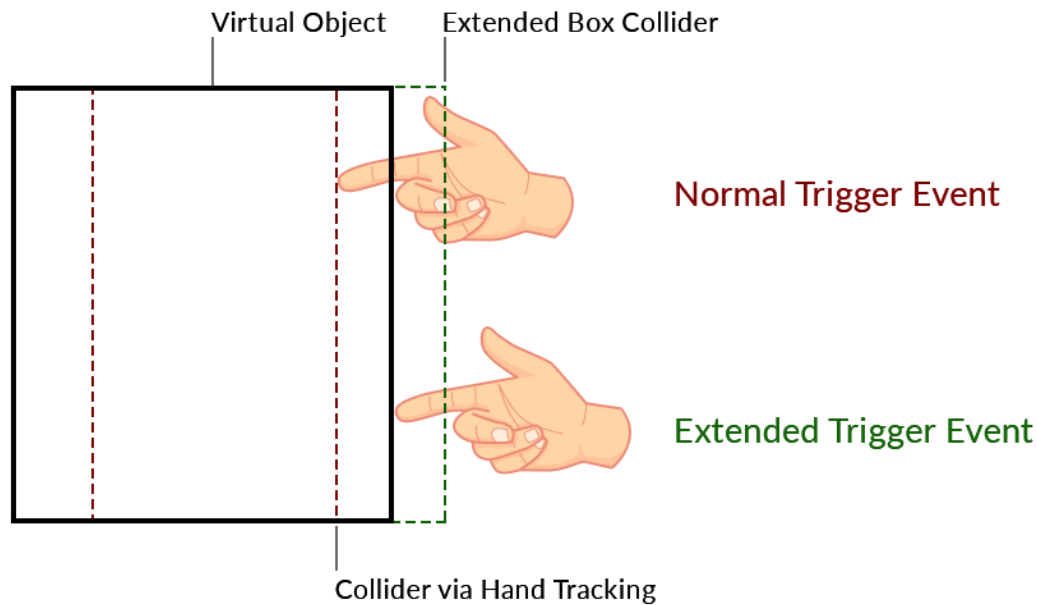


Figure 6.4 – A diagram displaying the effect of the extended box collider.

Figure 6.4 explains best how the incorrect margins of the Meta hand tracking are compensated by the *Collider* for the turn table. As displayed in the diagram, the extension of the *Box Collider* component of the object in Unity, is *virtually* approximately just as large as the error margin caused by the hand tracking of the Meta. Therefore, in practice, the touching of the turn table should feel more genuine and intuitive.

The second requirement for this iteration is to implement a selection procedure, so that the user is able to select a specific area of the garment to annotate. As the location of the annotation is an important factor, the more specific the selection procedure can be, the better. As the timespan for the project is relatively short, the attention span for the selection procedure should be sufficient, yet not too extensive. Hence, there was chosen to implement a ‘quick’ solution. The 3D model provided by PVH exists out of several parts, where each part represents a different area of the garment. For each area of the garment, a *Box* or *Mesh Collider* was made, allowing for detection of *Hand Triggers*. Each time a user’s hand *collides* with the *Colliders* of the parts of the garment, the *Trigger* event is called. Upon triggering of this event, the color of the part is changed, which makes the part highlighted and selected. To select another area, the user simply has to touch another area.

Thirdly, a basic tutorial should be implemented, explaining the interaction procedures of the application. The basic interaction procedures that will be explained are: rotation of the model, selection of an area, and the creation of an annotation. Each tutorial part exists out of a 2D-image, anchored to the *Meta Camera Rig*, which is a component in the Unity project. By anchoring the images to the *Camera Rig*, the images will remain visible when the orientation of the Meta 2 device changes. This makes it less likely that the user will miss any of the explanations provided in the tutorial.

6.4 Third iteration

The main goal of the third iteration is to implement the actual annotation recording system. Annotations should be stored in a file, so that these can be saved and loaded. Furthermore, as mentioned before, the selection procedure is critical and therefore it must be improved.

6.4.1 Iteration requirements

No.	Description
1	Improve the selection interaction to be more precise
2	Implement an interaction procedure to start a new recording or annotation
3	Implement the storing (saving and loading) of annotations
4	Implement the overview of annotations (with their correct positioning) on the garment

Table 6.5 – Requirements for the third iteration of the prototype.

6.4.2 Procedure

First, the goal of the last iteration was to make the selection interaction more precise. The selection interaction has completely been altered. After evaluation with Hecla, a new idea was introduced for the selection procedure: ‘pushpin’ interaction. The pushpin interaction included a ‘floating’ pushpin, which the user was able to drag and drop onto the garment, to be able to pinpoint the sought location. The precision of this new selection procedure is about 1 mm, meaning that the user is able to select an exact location for every millimeters of the garment.

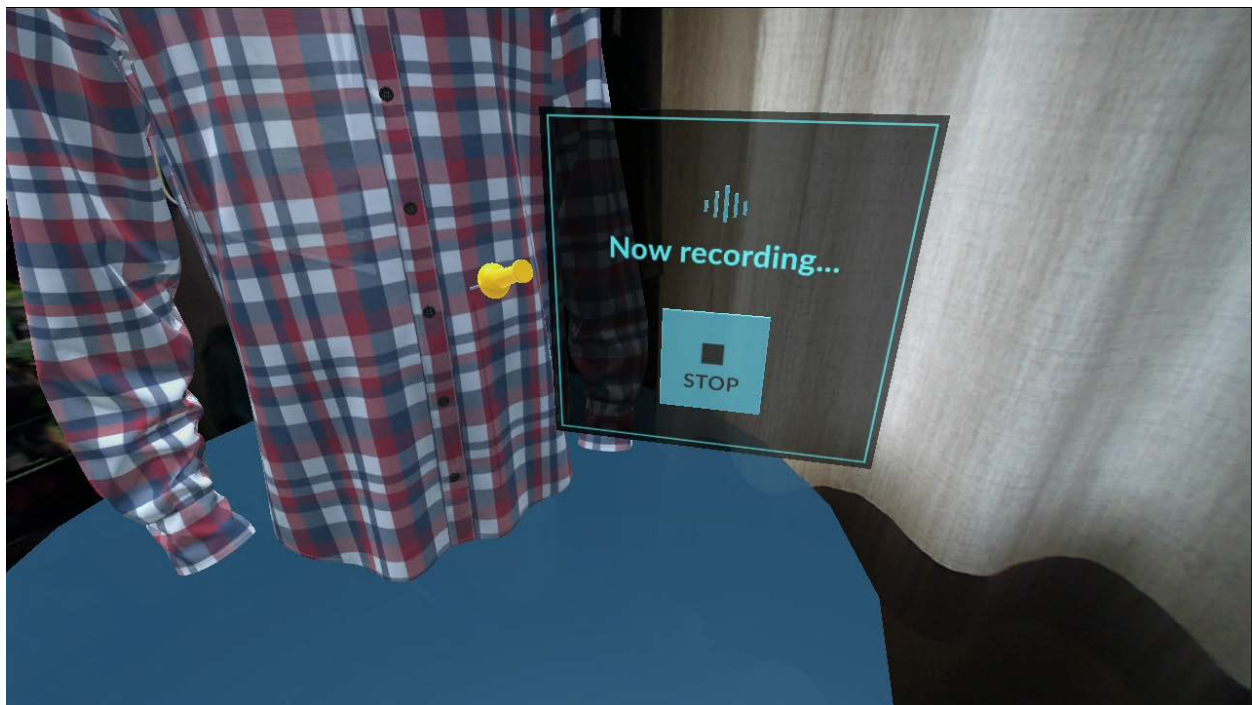


Figure 6.5 – An example of the 2D interface elements concerning the annotation.

Second, the interaction procedure to start a new recording or annotation has been implemented. The procedure for interacting with the pushpins existed out of 2D-interface elements, roughly the same size as the tutorial screens mentioned earlier. According to theory, 2D-interface elements are best suited for these forms of interaction, as stated ‘2-D interaction on a physical surface provides a sense of feedback that is especially useful for creating objects, writing, and annotating’ by Bowman et al. [36]. There are several actions required to start an annotation. First, the user has to *confirm* the position of the pushpin, so that the pushpin no longer can be moved. Then, the user has to *start* a recording of voice input and lastly the user has to *stop* this recording. When the recording has been stopped, the voice input will be saved. In order to match the pushpin location to the recording, both the pushpin data and the voice recording will be saved using the same *InstanceID*. The *InstanceID* of an object in Unity is always unique and will be the same (if instantiated in the same order) for every startup of the application. In figure 6.6-a to 6.6-d, the sequence of 2D interface elements is shown.

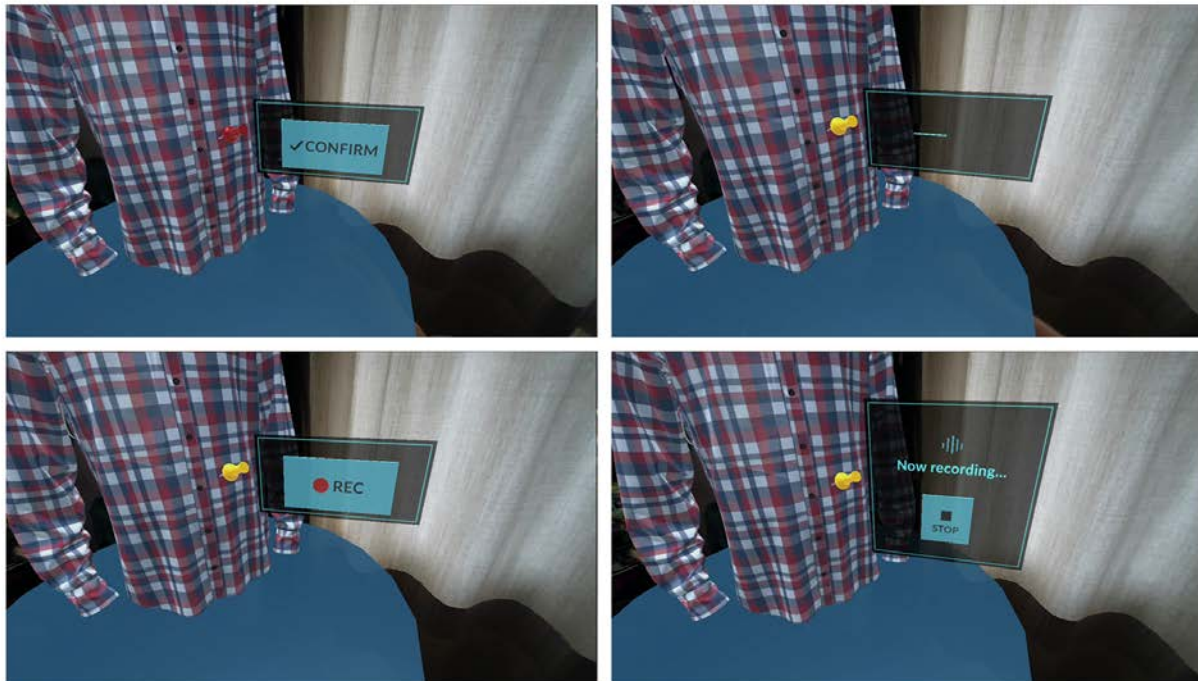


Figure 6.6 (a, b, c, d from top left to bottom right) –
a: Confirm button: confirms location of pushpin,
b: Loading screen: shown after each action,
c: Record button: starts new recording,
d: Status screen: displays the current status of the recording.

To interact with the 2D buttons shown in Figure 6.6, the user simply has to put his or her hand on the button. The button will then highlight yellow (the renderer displays this as green, due to the blue origin of the texture), as displayed in Figure 6.7.

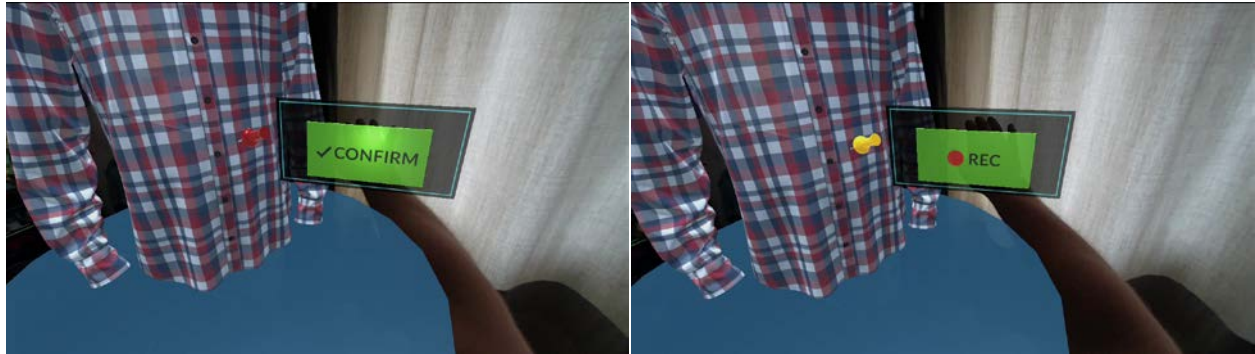


Figure 6.7 (a, b from left to right) – Highlighted buttons allow for visual feedback.

After the user has created a new annotation, the annotation can be replayed when hovering over the pushpin by hand. This is displayed in Figure 6.8.



Figure 6.8 (a, b, c from left to right) –
a: Play button: replays the recording of the annotation,
b: Loading screen: displayed between each action,
c: Status screen: displays the current status of the recording.

Continuing, the storing and loading of annotations has been implemented in the application as well. Every annotation gets saved with the following information: *location*, *id* and *instanceID*, where the last mentioned aspect is a direct link to the recorded file (i.e. annotation with pushpin with *instanceID* 7503 corresponds with the recorded sound input of file 7503.wav). A list of annotations is updated and saved to a data file. With each added annotation, the data file is updated and overwritten with the new list.

Lastly, the overview of annotations has been added to the last iteration as well. When achieving requirement number 2 ('implementing an interaction procedure to start a new recording'), the last requirement should also have been met, and indeed it has. When adding a new recording or annotation, the pushpin is saved and displayed on the correct position on the garment. When the garment is rotated via the turn table, the pushpin rotates simultaneously, hereby preserving its correct position on the garment.

6.5 System structure

6.5.1 User interface

The user interface is one of the most important aspects of the AR application, as it mainly determines how a user interacts with the application. Combined with the interaction techniques, this determines the overall usage of the application. Over the course of Creative Technology, undersigned has gained a lot of experience with 2D, flat, user interface design. However, the rules for interface design in a 3D environment are completely different.

6.5.1.1 Environment

There are several categories in which the user interface can be divided. Firstly, there is the 'environment' category, which covers the visuals of the 3D garment model and its environment. Later in this report, the interaction with the 3D garment model will be discussed extensively. For the environment, a basic round platform was chosen to place the 3D model on. The platform was chosen to be round, as it suggest rotational movement, similar to a steering wheel, a round stool or a turn table. The 3D garment is attached to the turn table, technically, while visually it floats slightly above the table. The size of the garment was chosen to be as close as possible to the one-to-one ratio, with virtual representation one the one side and real-world size on the other.

6.5.1.2 Tutorial

Secondly, there is the 'tutorial' or 'introduction' category in which the application explains the basic interactions to the user. In most 2D interfaces, the knowledge base of the end user is extremely large, because of the currently large number of 2D interfaces in software. Unfortunately, the amount of immersive applications at the time of writing does not match the 'mainstream' level of interface design. For these reasons, a basic tutorial is required for the user to understand the basic principles of the application. The basic tutorial starts with a brief explanation about the rotation of the turn table, explained in the next paragraph.

6.5.1.3 Annotation

Thirdly, there is the 'annotation' category. As the name suggests, this category covers the interface design necessary to view, create and interact with the annotations. The interaction with the interface will be discussed later in this report.

6.5.2 User interaction

Having a virtual interface is one thing, *interacting* with the interface is another. For each interface element the intention and outcome should be clear. In other words: each interface element should have the correct affordance.

6.5.2.1 Tutorial

The same rule that applies for interface design in 2D versus 3D: everything is completely different. Henceforth, the interaction with the virtual elements in the 3D environment,

should be explained in the basic tutorial. As mentioned before, the first tutorial screen shows how to interact with the turntable, hereby helping users to rotate the table and thus rotating the garment. The tutorial screen shows up for a number of seconds, so that it is ensured that the user was able to actually read and comprehend the brief explanation on the screen.

After the user has seen the first introduction screen, the environment will load the 3D objects. The user can now see the table (top of the table) and the 3D garment. Since the tutorial text gave a hint to try out the rotation, the first thing the user will try to do is to rotate the turn table. Having successfully rotated the table and the garment, the application will show the next tutorial screen (with a delay of 2 seconds).

The next step, required for the user to understand the interaction, is explaining how to handle the pushpin. The pushpin is stuck on the table and the user is able to grab and drop it to the garment, hereby placing a 'marker' on the location that the user wants to annotate. The pushpin explanation is slightly more difficult, hence it requires more tutorial screens. The first screen tells the user to grab the pushpin and drag it into the air. After the user has successfully done so, the second screen will explain to the user how to drag it onto the garment. Once again, if successful, the user should now see the interface buttons for the annotation.

6.5.2.2 Environment

The interaction with the objects within the AR environment is relatively straightforward. As mentioned before, the garment can only be rotated. In order to rotate the garment, the user has to place his or her hand in the round table-top. Once the application detects a successful interaction, the table-top will be highlighted. This visual feedback is extremely important, letting the user know that the touch is recognized. To rotate the table and the garment, the user now simply only has to move his or her hand to any direction. The rotation will start immediately, so that the user simultaneously sees the effect his or her movements is having on the environment.

6.5.2.3 Annotation

For this project, the actual annotating of the 3D garment is the most important interaction. A successful annotation requires the following information: location (x, y, z), a sound recording and possibly the raw drawing input. The location will be given by the user via the placement of the pushpin. The pushpin is located on top of the round table. In order to pick a location, the user can grab and drop the pushpin to the desired location on the garment. Once the pushpin is dropped, a UI-element pops up, containing the 'start recording' button. The user can press this button to start recording an annotation immediately.

6.6 Implementation

6.6.1 Software

The Meta 2 SDK (Software Development Kit) was used to develop the application. The Meta 2 SDK supports two major components: Unity and OpenVR. A brief explanation per major component is followed. Moreover, Microsoft Visual Studio was used to work in C#, where Adobe Photoshop – well-known photo editing software – and Adobe XD – relatively new user experience design software – were used for the 2D designs and assets of the application.

6.6.1.1 Unity

Unity is a cross-platform game-engine and game-development software developed by Unity Technologies. The programming language that is suited for Unity applications is C#. Creative Technology has offered several courses related to development with Unity. For this project, it has been chosen to work with Unity to develop the application, during the specification phase. Why Unity has been chosen, is mainly due to the support of the Meta 2 SDK. Since the Meta 2 SDK only supports two major components / software suites: Unity and OpenVR, the choice had to be made between Unity and OpenVR. Briefly said, the OpenVR system focuses more on VR application, wherefore the AR support is less extensive compared to the Unity system. Even so, undersigned has some experience in Unity and C#, while having no experience with OpenVR. Due to the timespan of this project, the extra time needed to ‘set up’ the OpenVR environment to work with the Meta 2 could not be permitted. Therefore, all in all, it was chosen to work with Unity.

6.6.1.2 OpenVR

The other major component supported by the Meta 2 SDK is OpenVR. OpenVR is an SDK and API (Application Programming Interface) developed by Valve, mainly with the goal to support HTC Vive development, while it also supports other XR (X Reality / Cross Reality) devices. OpenVR provides an API with plugins to support integration with Unity and Unreal Engine. For this project, OpenVR has not been used, as the support for OpenVR is mainly focused on VR systems rather than AR systems.

6.6.1.3 Adobe XD

Adobe XD is a user experience design program, suited to create UX-related designs ranging from wireframes to interactive 2D prototypes [37]. For this project, the program was mainly used for *wireframing* – creating skeletal frameworks for the interface. All wireframes exist out of basic black-and-white designs, merely showing the conceptual placement and size of each interface element.

6.6.1.4 Adobe Photoshop

Adobe Photoshop is a graphic editor or photo editing program developed and published by Adobe Systems, in 1988, suited for Windows and macOS [38]. The program was used for this project for creation of the image assets for the application, such as the 2D interface elements

of the tutorial. The wireframes created in Adobe XD are ‘beautified’ via means of Photoshop, to allow for implementation in Unity.

6.6.2 Hardware

As pointed out earlier, the prototype makes use of the Meta 2 device. The Meta 2 is superior to other AR devices such as the HoloLens and the Magic Leap, because of its extremely wide FOV (field of view) of 90° combined with a 2560 x 1440 high-dpi display, offering the most immersive experience of the market at the time of writing.



Figure 6.9 – A screenshot from one of the demo-videos found on the website of the Meta Co.

6.6.3 Hardware components

The Meta 2 is a fairly advanced device, existing out of many components. A list of all the components of the Meta 2 device is shown in Table 4, related to Figure 6.10.

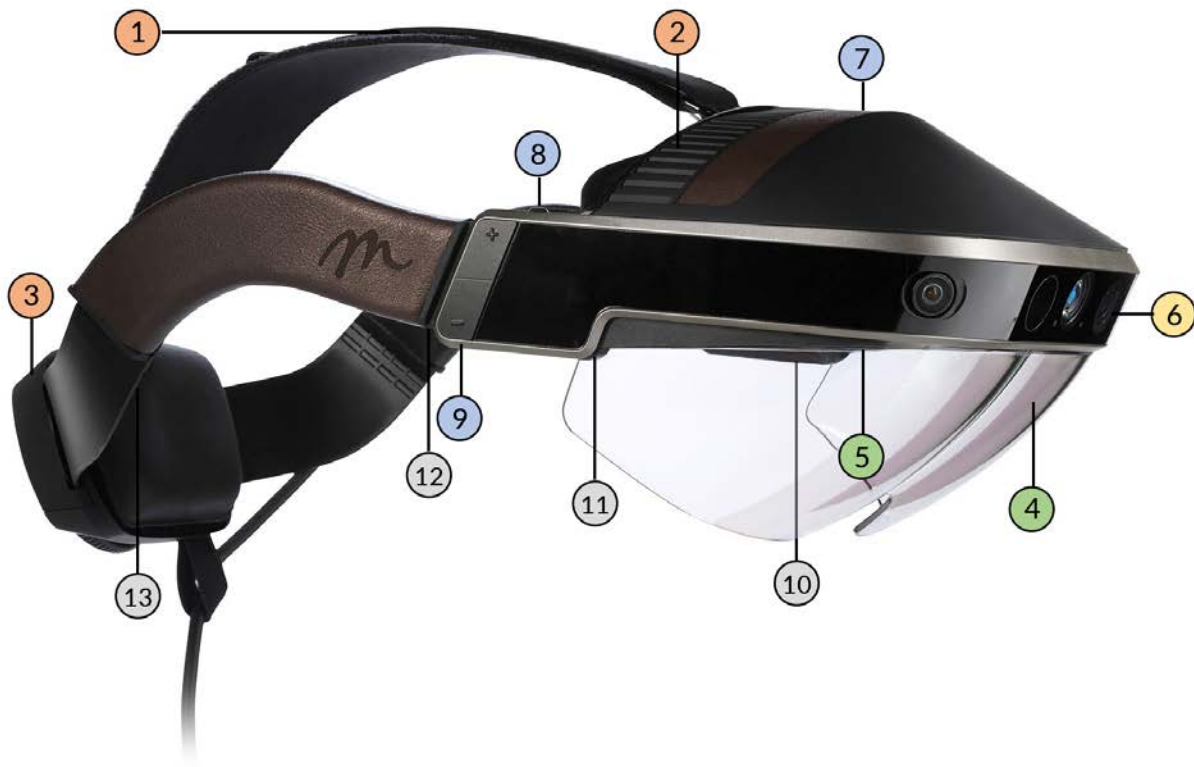


Figure 6.10 - Side-view of the Meta 2 augmented reality goggles.

No.	Type	Description
1	Comfort	Top strap
2		Forehead support
3		Adjustment Dial
4	Vision	Visor
5		Display
6	Technical	Array of sensors
7	Input	Display power button
8		Snapshot button
9		Volume controls
10	Sound	Front left speaker
11		Front right speaker
12		Rear left speaker
13		Rear right speaker

Table 6.7 – Legend of the Meta 2 side-view shown in figure 6.10.

7. Evaluation

During the evaluation phase of the project, a critical look is taken at all of the resulted prototypes. Whether or not all the requirements set in the specification phase are met, is discussed at ‘functional testing’. The final evaluation part exists out of user evaluations, where the added value of the prototype is examined in a ‘real’ scenario.

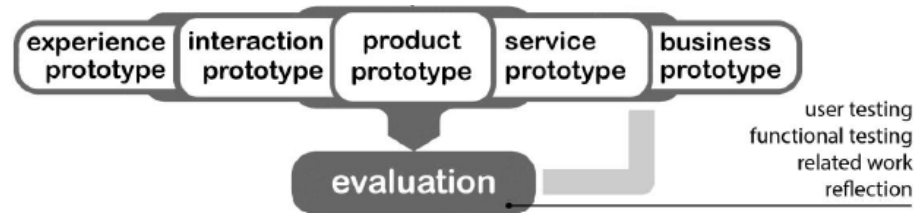


Figure 7.1 – The diagram for the ‘evaluation’ phase found in ‘Design Methods for Creative Technology’ by Mader and Eggink [30].

7.1 First prototype

To proceed with the second iteration, the first iteration had to be evaluated, so that improvements for the first iteration can take place during the second iteration. In table 6.2 is stated whether or not the requirements set in the beginning of this iteration have been met. A brief explanation per requirement is mentioned below.

No.	Description	Achieved
1	Set up the Meta 2 device to work properly with Windows 10 and Unity	Yes
2	Create an application in Unity that allows to see objects through the Meta 2	Yes
3	Implement a 3D-model of a garment designed by PVH so that all visuals are similar to a real-life version of the garment	Yes
4	Implement an interaction to be able to move, rotate or resize the 3D garment	Partially

Table 7.1 – Evaluation of the requirements for the first iteration of the prototype

As displayed in table 7.1, three of the four requirements have been met. A brief discussion per requirement is followed. First of all, setting up the Meta 2 device to work with Windows 10 and Unity has been relatively fast and easy. Some computers showed issues with the Meta 2 driver causing instability when running any AR application. On the latest hardware, the device and its software should run smoothly and without error. Second, creating an application in Unity, connecting the Meta 2 SDK succeeded almost instantly as the Meta 2 assets for Unity provide this functionality. Third, visualising the 3D model of a garment succeeded as well. As a partly goal of the application is to ‘preview’ the garment design in its environment, the garment should be displayed as realistic as possible. The high resolution and high DPI display of the Meta 2 device is perfectly suited to do so. All of the seams and detailed textures of the shirt are displayed crisp and clear.

Lastly, the last requirement ‘To implement an interaction to be able to move, rotate or resize the 3D garment’ succeeded *partially*. Summing up the issues caused by the first chosen interaction ‘Grab Interaction’: stuttering, hard-to-use, unwanted triggers and two-handed. Considering these issues, another interaction method was chosen. The turn table interaction has fewer possibilities, meaning, it lacks the possibility to resize and/or move the object, while it did appear to be working significantly better and more intuitive. Reasons to why the turn table interaction was considered to be better over the grab interaction are:

- proved to be more stable
- only one ‘action’: rotating has less room for error
- user can walk closer and further away from the object to ‘zoom’

On the other hand, the turn table interaction does not allow for movement of the objects. In some cases, the most important objects such as the garment were not rendered in sight by the Meta 2. The user had to turn around to be able to see the objects, which is an unpleasant, unnecessary and unintuitive step. Some minor issues with the *Collider* component of the turn table seemed to cause unexpected behaviour, an issue to improve in the following iteration.

7.2 Second prototype

Evidently, to proceed with the third iteration, the second iteration has to be evaluated, so that improvements for the second iteration can take place during the third iteration. In table 6.4 is stated whether or not the requirements set in the beginning of this iteration have been met. A brief explanation per requirement is mentioned below.

No.	Description	Achieved
1	Improve the turn table interaction to exclude unexpected behavior	Yes
2	Implement a selection interaction in which the user can select a specific area of the garment	Partially
3	Implement a basic tutorial in the application to explain to the user how the interaction and the UI works	Yes

Table 7.2 – Evaluation of the requirements for the first iteration of the prototype

As displayed in table 7.2, two of the three requirements have been met. A brief discussion per requirement is followed. Firstly, the turn table interaction has been improved by extending the *Box Collider* of the turn table object. During the evaluation at Hecla, the improved *Box Collider* proved to be working as expected, the unexpected behavior was minimized. Therefore, the first requirement for the second iteration has been achieved.

Second, the selection interaction of the application has *partially* succeeded. Even though the selection procedure has been implemented, the accuracy of the selection interaction varies from 1 cm² to approximately 20 cm². Since the accuracy of the selection of an area depends on the separate areas of the garment, there is no constant accuracy. If the user wanted to select, for example, the top left corner of an area of the garment, the selection procedure of

this iteration did not allow for this, as the only possibility was to select the entire area of the garment.

Third, a basic tutorial was implemented, with a goal of explaining the interaction process to the user. Each tutorial screen did its explaining relatively well: the users seemed to understand the interaction process much better after the brief explanation. However, there were some issues with the first tutorial screen: the fixed amount of seconds to which the screen was visible (the amount of seconds before the screen was set to be hidden) seemed to conflict with the calibration process of the Meta 2. For example, if the Meta 2 was calibrating the environment, the Meta 2 would display a screen showing ‘move your head from left to right’, where this screen would overlap the tutorial screen. This would mean that if the calibration process of the Meta 2 would take x seconds, the tutorial screen would be visible for $(n - x)$ seconds, making it difficult for the users to read the information displayed on the tutorial screen.

7.3 Final prototype

To evaluate the status of the final prototype and hereby to determine whether or not the project has succeeded, the third iteration has to be evaluated. Apart from the user evaluation, which is described in full detail in the ‘Evaluation’ chapter, the functional requirements are stated in Table 7.3. Whether or not the requirements set in the beginning of this iteration have been met, is also displayed in the table. A brief explanation per requirement is mentioned below.

No.	Description	Achieved
1	Improve the selection interaction to be more precise	Yes
2	Implement an interaction procedure to start a new recording or annotation	Yes
3	Implement the storing (saving and loading) of annotations	Yes
4	Implement the overview of annotations (with their correct positioning) on the garment	Yes

Table 7.3 – Evaluation of the requirements for the third iteration of the prototype.

As displayed in Table 7.3, all requirements have been met for the third iteration of the prototype. A brief discussion per requirement is followed. First, the selection interaction method has been improved significantly. User tests showed that the accuracy of the new selection method is about 1 mm^2 , meaning that the user was able to select an area of the garment as specific as 1 mm by 1 mm . During the testing, the *Box Collider* issue similar to the turn-table interaction appeared to also have effect on the pushpins. A similar solution as displayed in Figure 6.4 was applied to solve this problem and prevent this unexpected behavior.

Second, the interaction procedure for starting a new recording has been implemented. The 2D interface elements with large buttons and clear, brief descriptions for each action behind every button performed well in the user tests. It was pointed out during evaluation that

having the correct, familiar icons for every action would increase the understandability and usability of the interface elements. Therefore, a red dot icon has been added to the 'record' button, a checkmark to the 'confirm' button, a square 'stop' icon to the 'stop' button, and so forth. This change did indeed increase the performance of the interface elements, usability-wise. The affordance of the buttons has been improved using the western symbols.

Third, the storing and loading of annotations has successfully been implemented. During debugging, the `Reset()` function was loaded every time to prevent issues. The `Reset()` function has written to delete the data file of the annotations and to clear the annotation list in the memory of the application, so that for every restart of the application there would be no annotations loaded. However, since this functionality was part of the requirements, the functionality has been built. Upon loading the application, all existing annotations were loaded via the annotation data file. For each loaded annotation, a pushpin object was instantiated into the application with the correct positioning. One remark is that separate save files is not yet been made possible, yet the program was coded so that this functionality can easily be added in the future.

7.4 Functional Testing

To best determine the status of the prototype, all requirements set in the specification phase are evaluated by the researcher. All the requirements that have been achieved strengthens the prototype, whereas the unmet requirements allow for discussion and possibly future work.

No.	Requirement description	Priority	Achieved
Functional requirements			
1	The user is able to wear the Meta 2 device and walk around properly in an area of at least 4 m ² .	Must	Yes
2	The user is able to read the instructions of the application	Must	Yes
3	The user is able to view the chosen garment through the application, in sufficient detail	Must	Yes
4	The user is able to rotate the chosen garment through simple turn-table interaction	Must	Yes
5	The user is able to walk closer and farther away from the garment, while the garment and table remain in the same place	Must	Yes
6	The user is able to select a specific area of the garment	Must	Yes
7	The user is able to start a voice recording after having selected a specific area	Must	Yes
8	The user is able to control the recording (play, pause, stop) when wanted	Must	Yes
9	The user is able to open and view the newly added annotation on the 3D-model	Must	Yes
10	The user is able to go back to the overview	Must	Yes
11	The application contains 'grouping' functionality, where a relatively dense area of the garment with plenty of annotations is grouped into one larger pushpin	Should	No
12	The application uses speech-to-text software to summarize the annotations	Could	No
13	The user is able to filter the annotations based on subject or date	Could	No
14	The application allows to dynamically load a new 3D-garment with a separate set of annotation	Could	No

Table 7.4 - Evaluation of functional requirements

From the results of table 7.4, can be concluded that 11/14 of the requirements labeled with 'Must' are implemented in the prototype. During the specification phase, it has been specified that in the final prototype, all 'Must' and 'Should' requirements must be met for a complete successful outcome. Unfortunately, the one and only 'Should' requirement, requirement no. 11, has not been met. Therefore, it can be said, relatively straightforward and mathematically, that the project has been successful for 10/11 or 90,9%.

7.2 User evaluation

Non-functional requirements			
A	The user interface and interaction of the application should be clear and intuitive	Must	Yes
B	After the basic tutorial, the user should understand all the explained interactions properly	Must	Yes
C	The application should react to the users hands properly	Must	Partially
D	The user should be able to successfully create annotations	Must	Yes
E	The annotations should have clear sound recordings, with understandable speech	Should	Yes

Table 7.5 - Evaluation of non-functional requirements

As shown in Table 7.5, 4 out of 5 non-functional requirements have been met. All of the results displayed in Table 7.5, have been resulted from user testing. The main issue regarding requirement C, was due to the unreliability of the SLAM tracking methods of the Meta 2 device. Users found that the device was sometimes able to track their hands perfectly, whereas other times the device completely froze or did not detect hands properly. The *partially* conclusion can therefore be appointed to the software of the Meta 2 device, rather than the prototype.

7.3 Expert evaluation

Towards the end of the project, an expert evaluation was held at the office location of Hecla. Two experts in the field of audio-visual technology have tested the second iteration of the prototype, which lead to some promising results.

7.3.1 Structure of evaluation

The evaluation was performed in the form of a semi-structured interview. Several aspects of the prototype have been chosen to be evaluated. Evidently, the experts were given no explanation regarding usage of the application. The only explanation that had been done beforehand was a brief explanation regarding the usage of the Meta 2 device.

The aspects that were written down to be discussed have been displayed in Table 7.6.

Aspect	Importance
Visuals	Medium
Rotate action	High
Selection procedure	High
Tutorial	Low

Table 7.6 – To be discussed aspects during the expert evaluation

7.3.2 Results of evaluation

The expert evaluation provided several promising results. First, the results of the evaluation are discussed in general, followed by a brief discussion of the most important findings during the evaluation.

7.3.2.1 General

Several general comments have been made by the experts. One of the most common comments by the experts was to note the advantage of being able to see the environment. The comments referring to this advantage, however, can be slightly biased, as the expert evaluation has been combined with another application test, using the HTC Vive VR system.

Other notices regarding the Meta 2 application were related to the visuals of the application. Merely only positive comments have been made related to the visuals of the garment of the application. The experts noticed that the finer details of the garment such as sewing lines and high-resolution texture had been displayed as realistic as possible.

7.3.2.2 Rotate action: Box Collider turn table

The incorrect turn table behavior has been considered to be one of the most important findings of the expert evaluation. The lack of usability regarding the turn table interaction – or more precise: the inconsistencies in expected behavior, had led to a plentiful of unsuccessful actions. The expert users were more often unable to perform a ‘rotate’ action than they were able to.

During the evaluation, the *Box Collider* had slightly been tweaked to improve the turn table interaction and to neutralize the unexpected behavior. The spontaneous action to improve the interaction, during the evaluation process, immediately had shown to be successful. In the last prototype, the *Box Collider* of the turn table has been improved to try to cancel out the usability errors.

7.3.2.3 Selection procedure: Pushpin interaction

The last aspect that was discussed had been the selection procedure, where a user should be able to select a specific area of the garment. The experts were not able to test this procedure to the full extent, since the selection procedure had not yet been implemented to the full extent at the stage of evaluation. At the time of the expert evaluation, the selection procedure merely allowed selection (highlighting) of a specific *area* of the garment. An *area* could vary in size, and had been dependent on the specific 3D model. Meaning, if a 3D model would have existed out of 3 sub-parts, for example, two sleeves and the main body, the user would only have been able to select 3 areas, using this selection procedure.

However, the experts were able to suggest several improvements to be made for the selection procedure. One of the suggestions had been to implement a virtual ‘pointer’ on

the top of the hand of the user, allowing for specific selection. After a brief discussion of this newly thought-of selection procedure, was concluded that the error rate of unintentional selection-actions would be worrying.

Following, one of the experts suggested an improvement, based on the previous suggestion. The improvement was: to drag and drop a 'pointer'-like object onto the garment, to be able to select an area. In the final prototype, this interaction procedure has been worked out further into the 'pushpin interaction'.

7.4 Internal evaluation

Lastly, an internal evaluation was made to reflect on the process of the project and the result: the final prototype. The evaluation is named an 'internal' evaluation as it is done by the researcher of this project.

7.4.1 Project processes

Looking back, critically, on the processes of this project, there are several things to be concluded:

- start-up process was somewhat slow
- too much time was spend on the first prototype
- too little focus on the annotation aspect of the application

As mentioned above, first, the start-up process of this project was somewhat slow. In the beginning of this project, during the 'ideation' phase, when exploring the possibilities for this project, the goal of the project was not as clear as desired. The initial name for the project was 'Augmented Reality Meeting Rooms', where the focus was on AR collaboration over large distances.

However, after a first meeting with PVH, the focus of the project shifted from AR collaboration to develop an AR application that will support the processes within the fashion company of Tommy Hilfiger Europe. Unfortunately, this shift was only made after the first two weeks, meaning that those two weeks of research could have been more focused and relevant.

Second, while working on the first prototype, there was too much time spend on perfecting the visuals of the application. Despite its importance, there was much more work to do concerning the other iterations of the prototype. Moreover, the issue with the first iteration of the prototype was to find a proper 3D model of a garment. There were no *free* and *high quality* 3D models of professional garments available. Luckily, after a while, PVH provided a 3D model that was perfect for the prototype. Less time could have been spend on the first prototype, to be able to focus more on the *annotation* in the second and third iteration.

Last, there was too little focus on the annotation aspect of the application. Despite the three iterations, merely one of the iterations allows for a complete annotation functionality. If the

second iteration would have supported the complete annotation functionality as well, the evaluation for the second evaluation could result in a better annotation model for the last iteration, improving the overall usability and performance of the prototype and the quality of this research.

7.4.2 Final prototype

Reflecting on the final prototype of this project, evaluating its technical, functional and non-functional aspects, there are several aspects that stand out:

- augmented reality has its flaws
- interface performance / usability improves over time

One thing that was immediately noticeable during this project was that augmented reality is complicated. What is complicated, comes with errors, appeared to be valid for augmented reality. Unfortunately, there were many errors concerning the Meta 2 device, where either the SLAM (simultaneous localization and mapping) tracking immediately stopped working or the Meta 2 device crashed right after booting up the application.

A positive aspect of the project, was that the interface usability has a shallow learning curve, meaning that a user can learn most of the interactions with the interface rather quickly. When experimenting with the prototype, undersigned noticed that interaction with the interface was a lot easier over time. Happily, at the end of the project, working with the final prototype was as fluent as can be.

8. Conclusions

This section summarizes the findings of the report. To conclude, the answers of the research questions are described.

8.1 State of the Art

During the literature and background research, several things were concluded to be able to start the project. First, existing work for garment simulation allows for high-quality simulation of cloth using complex particle systems. However, as high quality real time cloth animation requires an extreme amount of processing power, there had to be made a choice: real time and low quality animation or a static high quality simulation. Since this project works with AR, meaning the application is rendered in real time, and the processing power is relatively low (due to its resources being spend on the AR tracking processes), there was chosen to work with a static, high quality clothing model.

Second, answering the first sub question of the research questions, there were found to be several interaction types for augmented reality: tracking-based and controller based. Even though there are sometimes vague lines between these two types, most of the interaction methods can be divided into either category. In the tracking-based interactions type there are several aspects that can be distinguished: fiducial, finger-tracking, hand-tracking or even head-tracking. For this project, hand-tracking has been applied.

8.2 Stakeholder identification and PACT analysis

A stakeholder identification and analyses has been performed to understand the environment and context of the project. The primary target group for this project has been identified: designers (reviewer and reviewee). The secondary target group for this project concerned the team leaders. As a result of this analysis, a PACT analysis was worked out, describing three usage scenarios for the project. Each scenario described a real-world usage of the prototype.

8.3 Ideation and specification

During the ideation phase, several ideas were worked out and evaluated. Many of the ideas were eliminated, or labeled as possible additions for future work. In the specification phase, more detailed design choices were made to give clear restrictions and requirements to the to-be-made prototype. It was decided to create an annotation application, allowing designers to annotate their existing 3D designs of garments. The goal of the application was to improve the performance over the designers. The requirements of the prototype were prioritized using the MoSCoW method.

8.4 Realization and evaluation

During the ‘realization’ phase, several prototypes have been made. Three iterations were made for the prototype, each iteration improving on the previous iteration. After each iteration, an evaluation was made with user testing and for the second iteration, experts have tested the application. The results of each evaluation were worked into the following iteration of the prototype. After evaluation with experts had been found that the ‘rotation’ action had been the most difficult, with a relatively high error rate. Moreover, the experts had suggested a new interaction type for the selection procedure. Following, the ‘rotation’ action has been improved in the second and final prototype and the suggested selection procedure has been implemented in the final prototype.

The final prototype achieved 10 out of the 11 requirements that were prioritized according to the MoSCoW method with ‘Must’ and ‘Should’. The requirement that was not included in the prototype was to group the annotations that were located nearby each other.

The first iteration of the prototype has been focused on setting up the Meta 2 and Unity project to work properly. A 3D model of a garment had been obtained from PVH, to display a piece of clothing as close to the proposed context of the application as possible. A simple interaction to manipulate the size and position of the 3D model had been implemented. After evaluation of the first iteration, the ‘manipulate’ or ‘rotate, resize or move’ actions had been categorized as ‘partially’ succeeded.

Following, the second iteration aimed to improve the requirements marked as ‘failed’ or as ‘partially succeeded’ in the first iteration. In addition, the requirements of the second prototype were extended with two new features: a selection procedure and a usage tutorial. The implemented selection procedure was fairly primitive, as followed from the evaluation. The ‘rotation’ action had been improved significantly, after the expert evaluation. A different *Box Collider* cancelled most of the unexpected behavior that was found during evaluation.

Lastly, the third and final iteration of the prototype focused on the annotation process itself. All steps required for a fully functioning annotation-system had been implemented: determining a position (selection), recording (voice input) and storing and loading (data file) the annotations. User evaluation showed that the annotation-system worked as had been expected. There were several tracking-related issues, comparable to the *Box Collider* issue of the turn table interaction, discussed earlier. Unfortunately, there had been no time to be able to test the final prototype with experts.

User testing showed that the interface of the final prototype worked intuitively. The learning curve of the pushpin interaction that was introduced in the third iteration of the prototype showed a slight learning curve, though the learning curve was relatively shallow. After user evaluation it has been pointed out that the visuals of the application were realistic, displaying realistic colors, where the size of the garment had appeared to be slightly too large. Moreover, several limitations in the SLAM tracking technology of the

Meta 2 resulted in unexpected errors and unwanted behavior of the application. Fortunately, all of the users succeeded in creating an annotation on the desired location.

8.5 Research questions

Although these evaluations were successful and the prototype seemed promising, much work is to be done to be able to implement the application in the existing situation of Tommy Hilfiger Europe or PVH. All necessities for implementation and professionalization are discussed in the ‘future work’ section.

“What methods of interaction are used by existing augmented reality installations?”

The ‘State of the Art’ section of this document describes the found interaction methods. Briefly said, there is an extensive amount of interaction methods developed and researched over the past decades. However, all of these interaction methods can be divided into two categories: tracking-based and controller-based. Simply put, tracking-based solutions tend to have calibration issues and are more easy to break under imperfect circumstances (lighting, faulty sensors), where controller-based solutions offer more precise interaction methods.

“Which interaction method(s) work best for annotation of 3D-models?”

Which interaction type is best suited for which purpose, can be concluded. If the focus of the purpose is on precision: using a controller-based interaction will most likely be the most successful. If the focus of the application lies with ‘ease of use’, it depends on the frequency and duration the users will use the application. If users will use the application frequently and for longer periods of time, then a controller-based interaction system will provide best results. If, however, the users will only use the system once, for a short duration, tracking-based interaction systems are proven to be more intuitive and easy to use – hence providing better results.

Altogether, the answered sub questions formulate an answer to the main research question of this project:

“How to best design an interaction model to allow annotation of 3D-models in an augmented reality environment?”

The contextual answer for this project would be to design a 3D user interface in Unity, using 2D elements, supporting the latest AR technology of the Meta 2 device, using hand-tracking and the interaction models described in the ‘realization’ section of this project. Whether or not this is the best solution for annotation in 3D models, has yet to be studied more extensively. This project has shown a solution for an annotation application supporting immersive AR technology, allowing for reviewing of a garment.

9. Future work

In this chapter, the possibilities regarding continuation on this project are discussed. Among these possibilities fall: a research to test the effects of implementation of this prototype into the design processes of Tommy Hilfiger Europe, new ideas to work out and further research options.

9.1 Implementation at Tommy Hilfiger Europe

The most important aspect of the future work section would be to test the effects of implementation at Tommy Hilfiger Europe. To be able to implement this prototype into the design processes of PVH or Tommy Hilfiger Europe, evidently, the final prototype has to be worked out into a first release. Currently, the prototype does not have sufficient functionalities to be implemented in the company processes of Tommy Hilfiger Europe. Whether or not the performance of the fashion designers actually improves after introduction of this project into the company's structure, has yet to be researched.

9.2 Addition of new features

Continuing on this project, the prototype can be extended with new features to improve and increase its functionality. Logically, all concept ideas described in the 'ideation' section can be placed here. A few examples, the most important ones, are listed and briefly explained below.

9.2.1 Collaborative annotating

As mentioned earlier in this project, one prime advantage of AR is to be able to have full contextual awareness, allowing for collaboration. A large addition to the functionalities of this prototype would be to add the possibility of collaborative annotating. In order to do so, there are several aspects that have to be studied:

- Allowing for collaboration between Meta 2 devices: same visual output, multiple input possibilities
- Combining the Meta 2 input in the Unity application: allowing for multiple user input

9.2.2 Drawing input per annotation

One of the requirements that was labeled as 'Could' after MoSCoW-prioritization, was to be able to *draw* in 3D space for every annotation. Besides voice input, this would be a great addition to improve the understanding of the annotation. To do so, drawing input has to be saved and mapped to the 3D environment, which can be a difficult task.

9.2.2 Summarizing, grouping and filtering annotations

When adding many annotations to one garment, the overview of the application of the final prototype can be unclear and chaotic. To improve this, annotations could be grouped, where multiple annotations on similar annotations could be replaced by a slightly larger pushpin, for example. When interacting with the larger pushpin, the prototype could 'zoom in' on the group, displaying all of the smaller, regular pushpins that were in the group.

Another addition to the prototype, one that is slightly more difficult to implement is to *summarize* the voice input of the annotations. By *summarizing*, in this case, is meant to transform the voice input into text, and consequently by some means summarizing the text. This would allow for a faster insight into the content of the annotation, without having to replay the entire voice recording.

After having summarized the annotations, new possibilities are opened up for further improvement. One of these possibilities is to add a filter possibility, where the user would be able to select a keyword to only display the annotations that are relevant to this keyword.

9.3 Professionalization

Before implementation at Tommy Hilfiger Europe, there are several things to be made more professional, before the first release of the application. A list of the most important aspects to be improved is:

- transforming storage system to a database
- creating a (web) interface allowing for submission and retrieval of 3D models and corresponding annotations
- refactoring the code
- adding tests for the code

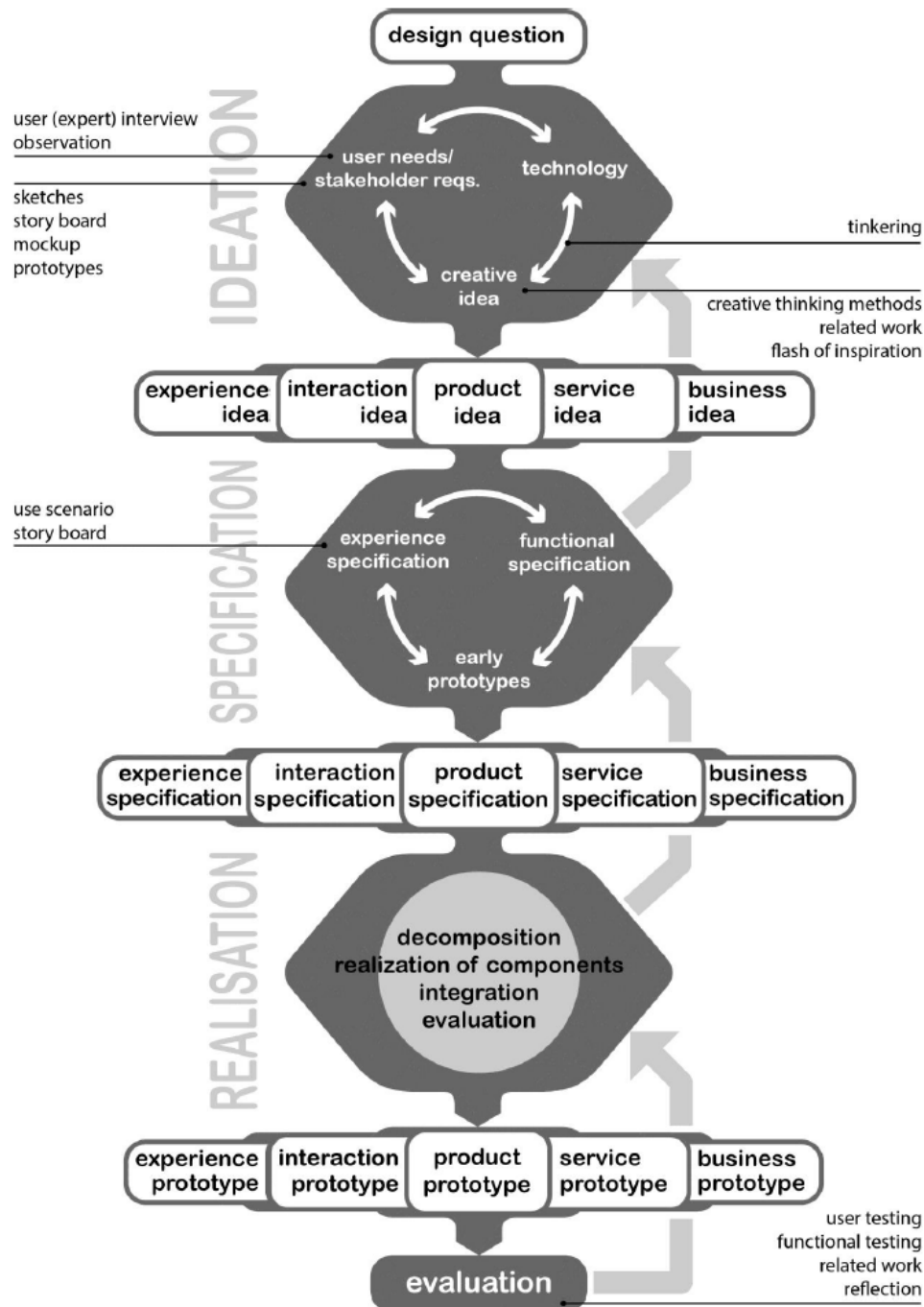
First, in order to improve and professionalize the storage system of the application, the storage system has to be transformed from a single file to work with a database. For this project, working with a database was overkill for the prototype. However, when dealing with great amounts of data, working with a database is necessary.

Second, in order to be able to work with the application, some sort of interface has to be developed, allowing for submission and retrieval of the 3D models and corresponding annotations. Currently, the prototype only works with one 'static' 3D model. To make the project work with multiple 3D models, the functionality to load a 3D model to annotate on has to be added as well.

Third and fourth, the code has to be tested. In order for the code to be tested, the code has to be refactored. Despite all of the *tidying up* attempts during the programming of this project, it is best to refactor the code to be more clear and understandable.

Appendices

Appendix A: Creative Technology Design Methodology



Appendix B: Semi-structured user test

This is one of the examples, briefly noted, of a user test template used for every iteration during the 'evaluation' phase.

Did you feel you had a decent representation of the garment, visually?

Yes, although the size maybe was a bit too big: visuals are good and high-quality. Colors are realistic.

Was the size of the virtual garment realistic, compared to what the size would be of the real produced garment?

Slightly too big: maybe a ratio of 1.1 compared to a real-life garment.

Did you understand how to manipulate the rotation of the garment? Did this work as intuitively as expected?

Yes, the tutorial explained this. It did work intuitively, despite sometimes not registering my hand.

Did you understand how to pinpoint an area to annotate the garment? Did this work as intuitively as expected?

Yes, the 'grab and drop me' message helped me understand. I understood what was required, unfortunately the grabbing of the pushpin did not work fluently.

Did you understand how to record an annotation, after selecting the location? Did this work as intuitively as expected?

Yes, I understood and it worked quite intuitively. Maybe adding sound-related icons such as a pause, play and record button can improve the understanding of the functionality.

Did you understand how to stop the recording of an annotation? Did this work as intuitively as expected?

Yes. Unfortunately, the opening of annotation was slightly troubling due to hand tracking issues.

Appendix C: Ethical Reflection report

Immersive Augmented Reality and Artificial Intelligence: an ethical review

A brief reflection on augmented reality, artificial intelligence and wearable computing

The subject of my graduation project for the bachelor study 'Creative Technology' at the University of Twente is, specifically stated '3D-model annotation for immersive Augmented Reality'. The client of the project is Hecla, an audio-visual integrator located in Hengelo, amongst other cities. However, the *actual* client for whom the project will be researched for is PVH (Philips Van-Heusen), one of the largest clients of Hecla and respectively one of the largest global apparel companies. Essentially, the final prototype of my graduation project will be made for fashion designers, allowing them to have a shorter feedback loop and to speed up the design process. Discovering the best way to annotate 3D-models in an Augmented Reality environment, opens up possibilities for new and more immersive forms of feedback than before.

The main technology that is relative to this project is Augmented Reality. 'Augmented Reality (AR) is a technology which allows computer generated virtual imagery to exactly overlay physical objects in real time' [4]. The technology allows developers to create new interactions between the digital world and the real world. Although the first AR-interfaces or systems were discussed nearly 40 years ago [10], it has become a hot topic over the past decade. Last year, AR was outlined to be in the 'through of disillusionment' column within 5 to 10 years, of the Gartner Hype Cycle for Emerging Technologies

[39]. This shows that the popularity has been incredibly large over the past few years, and thus the existing amount of research for AR has increased significantly.

There are different forms of AR. One of these forms has been adapted earlier by tech companies than the other. Looking at AR from a high-level, there can be two main forms distinguished. The first form being 'flat' augmented reality, projected by or on a screen, and the latter being 'immersive' or 'tangible' augmented reality.

The latter form of AR is advanced and requires systems that can be difficult to develop. Generally speaking, an immersive or tangible AR-set-up contains the following components: a HMD, HWD or HUD (head-mounted, head-worn -or heads-up-display), one or multiple tracking devices and a lot of processing power provided by a GPU (graphics processing unit) and CPU (central processing unit) [40]. There are several complete-package set-ups available for developers to buy and work with.

When having a complete set-up of a tangible AR-environment, the user is able to display any digital information (3D-models, data graphs, maps), on top of the real world. The power of augmented reality lies within finding the perfect combination between reality and digital augmentation.

To look at the ethical aspects of the graduation project, it's best to take one step back, and focus on the 'immersive augmented reality' technology, rather than focusing on annotation 3D-models via AR. There are several things to be considered of moral significance, considering immersive AR technologies.

Wearable Augmented Reality

Imagine wearable computing, combined with immersive AR, applied to our society to the extreme level. Depict every person on earth, walking around with augmented reality 'goggles'. The possibilities are endless and quite frightening.

Google has made the largest attempt in this field of immersive augmented reality in wearable of computing. The Google Glass, heads-up glasses for AR applications, were meant to be a big game-changer in the AR-domain and change the way people live. The heads up glasses were designed to be a pair of glasses, combined with a very small transparent display, overlaying the view of the user wearing the glasses. A camera located on the side of the glasses would be used as input for the device to be able to project digital information on the display in 3D-space.

The world was immediately skeptical when the Google Glass devices were announced and rolled out. There has been some research discussing the privacy concerns of the Google Glass project. S. Wagner has written a study discussing the potential drawbacks of the Google Glass [41]. In his study, he describes that such technological privacy harms are often hard to detect and to identify, as the general public often does not understand

the technology and its potential uses. The lack of understanding makes it difficult to understand whether the information that one is sharing, personal information, can be used inappropriately. The 'Big Brother is watching you' line from George Orwell [42] always comes to mind when thinking of companies such as Google and Facebook collecting all of our personal information.

The more information companies such as Google and Facebook retrieve from us, the more money they earn by selling this information. Conceptualizing having augmented reality 'vision' or 'goggles' via some head-mounted or heads-up display, implemented into people's everyday outfit, can change the way the world is perceived by people. Having a camera mounted on your head, a simple action such as taking a stroll through the park can be privacy-invading to other society members. Moreover, AR technology can take this privacy invasion several steps further. The new ways in which displaying digital data can be applied, such as real-time facial recognition used by the police in China, are 'Big Brother' worthy [43]. In China, the police has rolled out several tests with augmented reality glasses, used by police agents. The requirements over time of these AR-glasses was that they should be able to recognize a person within 3 seconds, allowing the police to quickly scan through their environment, noticing potential threads.

Conceptually speaking, over the years, artificial intelligence systems implemented into AR-glasses could eventually help us see the world differently. Evidently, AI technology was

developed in the first place, to make computers smarter in helping us, humans. Primarily, the overall improvements made by AI technologies in AR-glasses are developed with the objective to *improve* our lives. An example is help of A.I. in the kitchen, reminding us to take out the pasta on time, while cooking a specific recipe, which is also projected on our kitchen wall. Possibly, over time, the result of these will technologies will have the opposite impact: a negative impact. Most likely, we humans will become dependent of these technologies, trusting our own judgments less and less. To stick to the same example, over time, people will no longer know how to cook pasta properly, and always depend on their AR-glasses. More generally speaking, letting the software of the AR-glasses take simple decisions for us via AI, will be a potential downfall for our own decision-making authority and our own consciousness.

Augmented Reality as an persuasive technology

It is no surprise that Augmented Reality falls within the definition of persuasive technology, as the definition of persuasive computing is: 'the designing and use of technology with the specific intent of influencing or modifying behaviors, values or attitudes' [44][45]. Henceforth, all theories on ethics from the persuasive technology field can also be applied to AR.

To some extent, (almost) every existing AR application has some sort of persuasive design element in it. As simple as the persuasion can be, such as a hint to press a button when ready, the design does have the intention to alter the behavior of the end user. The persuasion is more powerful

too, compared to 'common' products, as the AR application can create an immersive experience, changing how we see and interact with the world around us [46].

Whether the persuasive design of the software in the AR goggles is ethically worrying, obviously depends on the software itself. However, it is the *potential* of the AR goggles combined with advanced software that is 'dangerous'. Immersive AR interfaces can potentially be extremely good persuaders: the best persuasive technology there is. The main reason for this near-perfect persuasion, is that the AR software simply can do things humans cannot do [46].

AR applications have the ability to utilize timing, user data, environmental data and ubiquitous contextual information to offer persuasive technologies at the right time and place, placed in the correct context of the end users' environment. Of course, these statements are merely a future prediction, and extensive research and development is needed before the above-mentioned level of software is reached.

In the end, potential manipulation of user's behavior by AR software is an ethical concern and software developers should keep this in mind. Then again, on the other side, are we not manipulated to act accordingly by the government and our environment already?

Google Glass

An example of the way the persuasive design is implemented into AR-technology is the 'Google Glass'. Back in 2013, Google

(or Alphabet) released the 'Google Glass', also known as 'Project Glass'. The wearable device developed by Google existed out of an optical head-mounted display (OHMD). Images were projected on a small area of the glass via a prism projector. Everybody wearing the Google Glass was able to interact with the wearable device via voice input. Example commands for the Google Glass were: 'ok glass, find a recipe for ...', 'ok glass, show me the news' and 'ok glass, record a video'. The 'ok glass' command was similar to the often-used 'ok Google' command for Android smartphones, as it was meant to trigger the device to listen to the user.

Though the project by Google was seemingly perfectly-fitting for a decent Sci-Fi movie, one could argue that the world was not ready for this device. Everybody that would be wearing the Google Glass, would be able to record everything they were seeing. Basically, Google gave the world the opportunity to everyone to wear a spy-cam and record the world unknowingly. Despite all its advantages and potential due to the possibility for fast and smart interactions, the project became a flop. After one restaurant went as far as to forbid the Google Glass [47], many other public spaces or catering companies started to follow this trend [48].

Other than recording the environment, without the environment noticing the event of recording, Google Glass and various immersive AR-projects provide several other aspects that can lead to an ethical discussion. Technology trends such as Artificial Intelligence via neural networks and machine learning,

Ubiquitous Computing and Wearable Devices, can all be summarized as 'smart' technologies. The adjective 'smart' can mean several things here, to clarify, 'smart' technologies are aware of context, autonomous (to some extent) and can operate interactively (to some extent). Some of these technologies are smart enough for us humans to communicate with, *verbally*. As the Google Glass works via voice-commands, the device is capable of understanding human language, meaning it can hear and understand, to some degree, what the user is saying. It is comprehensible that to some extent, users are uncomfortable with having a 'device' or 'digital assistant', understanding the audial and even visual context of their individual. The possibilities of Google using this data for *profiling*, which is one of the most important values where large tech companies such as Google and even more so, Facebook, owe a large share of their profit to [49].

Google Duplex

The intelligent (silent) voice from the Glass Project was something Google had been working on for quite a while. Similar assistants have been out there for quite a while, such as Apple's Siri, Microsoft's Cortana and Amazon's Alexa. The Google Assistant, a virtual assistant developed by Google that is available on their smartphone and wearable-device OS Android and on their smart home devices, debuted in May 2016 via Google's messaging app Allo. Only later, in February 2017, the virtual assistant was deployed on other Android devices. In the latest, annual I/O developers' conference on May 8,

2018, Google previewed a new feature called *Google Duplex*.

Google Duplex is an extension of the Google Assistant that allows the assistant to carry out natural, human-like conversations by mimicking a human voice. During the I/O developers conference of 2018 the Google Assistant performed a task of calling an actual hair salon to arrange an appointment, pretending to be as human as possible. The audience of the conferences started applauding, presumably since it was quite an astonishing performance by the Google Assistant. People were not sure how to react to this, whether to be scared, uneasy or to be impressed and wondered.

Many concerns were made over the societal and ethical questions that the Duplex technology raises [50]. Most of the concerns were having to do with humans, on the other side of the line, would not be able to tell the difference between the Duplex voice and an actual human being. This would mean that Google's Duplex would most likely pass the Turing Test [51]. Potentially, this milestone of a computer being able to mimic a human being is a large threat to societal safety. With false intent, computers can pretend to be a person, hereby acting on behalf of that person with great credibility. What if a family member's voice would be replicated, in order to gain access to personal information? The software has great potential for scammers, fake news and identity thefts.

Of course, the software was not designed with those false intentions in mind, however, these possible outcomes should

be considered. A lot of moral questions arise regarding this matter. Is it unethical for computers to mimic persons? The aim of Google Duplex is to act as an assistant, so therefore identity theft can be ruled out. However, will it be morally accepted to have to have a conversation with a 'robot', unaware of the fact that there is not actually another person on the other side of the talk?

Conclusion

All in all, immersive AR technology provides an extensive array of possibilities that are leaning towards possible immoral or unethical applications. In combination with advanced artificial intelligence, the potential threat of these applications to our privacy and safety increase. Once popular amongst the population, AR-goggles have great potential for governmental monitoring purposes, increasing the amount of available cameras for monitoring networks. On the other hand, the technology provides plenty of possibilities to make the lives of the end users better, aiding people with their everyday tasks.

Advanced artificial intelligence such as Google Duplex can show worrying scenarios, such as a 'robot' or virtual assistant mimicking a person's voice. To what extent this should be morally accepted, is to be discussed and made clear in the future. Whether or not the positive effects such as the ability to delegate more advanced tasks, which normally require the thinking power and the verbal skills of a human being, outweigh the negative effects of possible scamming or identity theft depends on where our society will draw the line.

Appendix D: C# code

MainController.cs

```
public void loadAnnotations()
{
    Debug.Log("Starting loading of annotations");
    SaveLoad.Load();
    SaveLoad.Init();
    for (int i = 0; i < SaveLoad.annotations.Count; ++i)
    {
        Debug.Log(SaveLoad.annotations.Count.ToString() + " annotations loaded.");
        Annotation d = SaveLoad.annotations[i];
        var spawnedAnnotation = Instantiate(annotationObject, new
Vector3(d.positionX, d.positionY, d.positionZ), Quaternion.identity);
        spawnedAnnotation.gameObject.tag = "PushPinContainer";
        spawnedAnnotation.transform.parent =
GameObject.FindGameObjectWithTag("PushPinParent").transform;
        spawnedAnnotation.transform.position = new Vector3(d.positionX, d.positionY,
d.positionZ);
    }
}

public void spawnPushpin()
{
    Vector3 position = new Vector3(0.0f, 0.0f, 0.0f);
    var spawnedAnnotation = Instantiate(annotationObject, position,
Quaternion.identity);
    spawnedAnnotation.gameObject.tag = "PushPinContainer";
    spawnedAnnotation.transform.parent =
GameObject.FindGameObjectWithTag("PushPinParent").transform;
}

public void playSuccessSound()
{
    successSound.Play();
}
}
```

AnnotationController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Meta;

public class AnnotationController : MonoBehaviour {

    public GameObject HUD_confirmPosition;
    public GameObject HUD_startRecording;
    public GameObject HUD_currentlyRecording;
    public GameObject HUD_finishedRecording;
    public GameObject HUD_playRecording;
    public GameObject HUD_playingRecording;
    public GameObject HUD_loadingScreen;

    AudioClip recording;

    string recordingFileName;
    AudioSource recordingToPlay;

    public GrabInteraction grabInteractionComponent;

    private bool inRecording;
    public struct annotationStruct
    {
        public GameObject pushpin;
        public GameObject pushpinPIN;
    }
    private annotationStruct activeAnnotation = new annotationStruct();
    Color standardPushPinColor;

    // Use this for initialization
    void Start () {
        // Set everything to null
        activeAnnotation.pushpin = null;
        activeAnnotation.pushpinPIN = null;
        inRecording = false;

        playRecording();
    }

    // Update is called once per frame
    void Update () {
        if(recordingToPlay != null && recordingToPlay.clip != null)
        {
            if (!recordingToPlay.isPlaying && recordingToPlay.clip.loadState ==
AudioDataLoadState.Loaded)
            {
                recordingToPlay.Play();
            }
        }
    }
}
```

```

public void StartNewAnnotation(GameObject annotationLocation)
{
    // Set new location of annotation and annotation pin to be active
    Debug.Log("Updated 'currentAnnotationLocation'");
    activeAnnotation.pushpin = annotationLocation;
    Debug.Log("Current annotation location: " + activeAnnotation.pushpin.name);
    activeAnnotation.pushpinPIN =
annotationLocation.transform.Find("PIN").gameObject;
    Debug.Log("Current annotation locationPIN: " + activeAnnotation.pushpinPIN.name);

    // Spawn the HUD to confirm positioning
    Debug.Log("Current annotation location child: " +
annotationLocation.transform.childCount.ToString());
    SpawnHUD(annotationLocation, HUD_confirmPosition);
}

public void SpawnHUD(GameObject location, GameObject spawnObject, float destroyAfter
= 0.0f)
{
    // Hide all other HUDs first
    DestroyAllHUDS();

    // SpawnPosition is slightly to the right of the objects position
    var spawnPosition = new Vector3(location.transform.position.x,
location.transform.position.y, location.transform.position.z - 0.25f);
    // Instantiate the object at the correct location
    var spawnedHUD = Instantiate(spawnObject, spawnPosition, Quaternion.identity);
    // Set the parent to be the parent of the location -- pushpinContainer
    var spawnParent = location.transform.parent;
    spawnedHUD.transform.parent = spawnParent.transform;

    // Make sure the positioning is correct
    spawnedHUD.transform.position = spawnPosition;

    // Activate the object in the hierarchy
    Debug.Log("Spawning new " + spawnObject.name + "HUD.");
    spawnedHUD.SetActive(true);

    // Boolean determines whether or not the object will be destroyed in 5 seconds
    if(destroyAfter > 0.0f)
    {
        Destroy(spawnedHUD, destroyAfter);
    }
}

public void SpawnHUDAfterLoading(GameObject location, GameObject spawnObject)
{
    SpawnHUD(location, HUD_loadingScreen, 2f);
    StartCoroutine(SpawnWithDelay(location, spawnObject));
}

IEnumerator SpawnWithDelay(GameObject location, GameObject spawnObject, float
spawnDelay = 2f)
{
    yield return new WaitForSeconds(spawnDelay);
    SpawnHUD(location, spawnObject);
}

```

```

public void DestroyAllHUDS()
{
    DestroyGameObjectsWithTag("Garment-HUD");
}

public void DestroyGameObjectsWithTag(string tag)
{
    // Find GameObjects with a given tag, and destroy them if found
    var previousConfirmHUDs = GameObject.FindGameObjectsWithTag(tag);
    if (previousConfirmHUDs.Length > 0)
    {
        Debug.Log("Deleting " + previousConfirmHUDs.Length + " GameObjects with tag: '" + tag + "'");
        foreach (var previousConfirmHUD in previousConfirmHUDs)
        {
            GameObject.Destroy(previousConfirmHUD);
        }
    }
}

public void ConfirmAnnotationLocation()
{
    // Confirm the location of the pushpin to be able to start annotating
    Debug.Log("Confirming annotation notation");

    // Set the Pin color to yellow, to be active
    SetPinColor(activeAnnotation.pushpin, Color.yellow);

    // Remove the grab interaction: moving of the pushpin is no longer allowed
    grabInteractionComponent =
activeAnnotation.pushpin.GetComponent<GrabInteraction>();
    Destroy(grabInteractionComponent);

    // Spawn the HUD displaying 'Ready to start recording'
    SpawnHUDAfterLoading(activeAnnotation.pushpin, HUD_startRecording);
}

public void RespawnMainPushPin(GameObject pushpin, GameObject pushpinContainer)
{
    var spawnPosition = new Vector3(0.1909998f, -0.04900156f, -0.2259569f);
    var spawnedObj = Instantiate(pushpin, spawnPosition, Quaternion.identity);
    spawnedObj.transform.parent = pushpinContainer.transform.parent;
}

```



```

public void hideAllNonActivePushPins()
{
    // Check if there are non-active pushpins, and hide them from the hierarchy
    var foundPushpins = GameObject.FindGameObjectsWithTag("PushPinContainer");
    if(foundPushpins.Length > 0)
    {
        foreach (GameObject foundPushpin in foundPushpins)
        {
            Color pinColor =
GetPinColor(foundPushpin.transform.Find("pushpin").gameObject);
            // If the Pin color of the pushpin is different than normal, it is
active
            if(pinColor != Color.yellow && pinColor != Color.green)
            {
                // Only set to disabled when non-active
                standardPushPinColor = pinColor;
                Debug.Log("Setting pushpin to inactive");
                foundPushpin.SetActive(false);
            }
        }
    }
}

public void showAndResetAllPushPins()
{
    // Restore the original overview of pushpins, setting everything to active
    var foundPushpins = GameObject.FindGameObjectsWithTag("PushPinContainer");
    if (foundPushpins.Length > 0)
    {
        Debug.Log("Setting all pushpins to active.");
        foreach (GameObject foundPushpin in foundPushpins)
        {
            if (standardPushPinColor != null)
            {
                // Setting all the colors of every pushpin to the regular color
                SetPinColor(foundPushpin.transform.Find("pushpin").gameObject,
standardPushPinColor);
            }
            Debug.Log("Setting pushpin to active.");
            foundPushpin.SetActive(true);
        }
    }
}

```

```

public void startRecording()
{
    // Recording has started
    inRecording = true;

    // Hide all other, non-active push pins
    hideAllNonActivePushPins();

    // Display 'recording...' HUD
    SpawnHUDAfterLoading(activeAnnotation.pushpin, HUD_currentlyRecording);

    foreach (string device in Microphone.devices)
    {
        Debug.Log("Name: " + device);
    }

    // Start recording
    recording = Microphone.Start("Headset Microphone (HyperX 7.1 Audio)", false, 10,
44100);
}

public void stopRecording()
{
    // Recording has now stopped
    inRecording = false;
    Debug.Log("Stopping recording!");

    // All pushpins can be shown again
    showAndResetAllPushPins();

    // Save recorded voice to file
    SavWav.Save(activeAnnotation.pushpin.GetInstanceID().ToString(), recording);

    // Save annotation list to save file
    SaveLoad.Save(activeAnnotation.pushpin);

    // Spawn the HUD to display 'recording finished'
    SpawnHUD(activeAnnotation.pushpin, HUD_finishedRecording, 5f);
}

public void playRecording()
{
    if(activeAnnotation.pushpin != null)
    {
        recordingFileName = "file:/// " + Application.persistentDataPath + "/" +
activeAnnotation.pushpin.GetInstanceID().ToString() + ".wav";
        Debug.Log("Playing sound: " + recordingFileName);
        StartCoroutine(playSound());

        SpawnHUDAfterLoading(activeAnnotation.pushpin, HUD_playingRecording);
    }
}

```

```

public void stopPlayback()
{
    DestroyAllHUDS();
    recordingToPlay.clip = null;
}

private IEnumerator playSound()
{
    recordingToPlay = GetComponent();
    using (var www = new WWW(recordingFileName))
    {
        yield return www;
        recordingToPlay.clip = www.GetAudioClip();
    }
}

public void openAnnotation(GameObject annotationPushpin)
{
    // Check if the annotation is in the saved list
    foreach (Annotation annotation in SaveLoad.annotations)
    {
        // If it matches
        if (annotation.id == annotationPushpin.GetInstanceID())
        {
            // Set the opened pushpin to be active
            activeAnnotation.pushpin = annotationPushpin;
            activeAnnotation.pushpinPIN =
annotationPushpin.transform.Find("PIN").gameObject;

            // Hide all other pushpins
            hideAllNonActivePushPins();

            // Annotation exists, open annotation
            // Display 'play recording' HUD
            SpawnHUD(annotationPushpin, HUD_playRecording, 5f);
        }
    }
}

private Color GetPinColor(GameObject pushpin)
{
    // Retrieve the PIN color from the gameobject named 'PIN'
    GameObject pin = pushpin.transform.Find("PIN").gameObject;
    return pin.GetComponent<MeshRenderer>().material.color;
}

private void SetPinColor(GameObject pushpin, Color c)
{
    // Set the PIN color for the gameobject named 'PIN'
    GameObject pin = pushpin.transform.Find("PIN").gameObject;
    Material m = pin.GetComponent<MeshRenderer>().material;
    m.color = c;
}
}

```

HUDController.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class HUDController : MonoBehaviour {

    public GameObject HUD_0_Rotation;
    public GameObject HUD_1_Selection;

    public GameObject garment;

    private int count;

    private bool hasSeenFirstScreen = true;
    private bool hasSeenSecondScreen = false;

    // Use this for initialization
    void Start () {
        count = 0;
        //SpawnHUD(HUD_0_Rotation, 12f);
    }

    // Update is called once per frame
    void Update () {

    }

    public void ShowTutorial(GameObject spawnObject)
    {
        // Only show tutorial if not already seen
        if(count < 2)
        {
            StartCoroutine(SpawnWithDelay(spawnObject));
        }
    }

    public void SpawnHUD(GameObject spawnObject, float destroyAfter = 0.0f)
    {
        // Hide all other HUDs first
        Debug.Log("Hiding all HUD elements");
        DestroyAllHUDS();

        Debug.Log("Hiding the garment and all pushpins");
        HideGarmentAndPushPins();

        // SpawnPosition is slightly to the right of the objects position
        var spawnPosition = new Vector3(0f, -0.133f, 0.689f);
        // Instantiate the object at the correct location
        var spawnedHUD = Instantiate(spawnObject, spawnPosition, Quaternion.identity);
        // Set the parent to be the HUD parent
        spawnedHUD.transform.parent = GameObject.FindGameObjectWithTag("HUD").transform;

        // Activate the object in the hierarchy
        Debug.Log("Spawning new " + spawnObject.name + "HUD.");
        spawnedHUD.SetActive(true);

        count++;
    }
}
```

```

        // Boolean determines whether or not the object will be destroyed in 5 seconds
        if (destroyAfter > 0.0f)
        {
            Debug.Log(spawnedHUD.name + " will be destroyed after " +
destroyAfter.ToString() + " seconds.");
            Destroy(spawnedHUD, destroyAfter);
            StartCoroutine>ShowGarmentAndPushPins(destroyAfter));
        }
    }

    public void DestroyAllHUDS()
    {
        DestroyGameObjectsWithTag("Garment-HUD");
    }

    public void HideGarmentAndPushPins()
    {
        var objects = GameObject.FindGameObjectsWithTag("ShowAfterIntro");
        foreach(GameObject obj in objects)
        {
            StartCoroutine(RespawnObjectWithDelay(obj));
            Debug.Log("Setting object: " + obj.name + "to inactive.");
            obj.SetActive(false);
        }
    }

    IEnumerator ShowGarmentAndPushPins(float spawnDelay)
    {
        yield return new WaitForSeconds(spawnDelay);
        Debug.Log("Setting object: " + garment.name + " to active!");
        garment.SetActive(true);
    }

    IEnumerator SpawnWithDelay(GameObject spawnObject)
    {
        yield return new WaitForSeconds(2);
        Debug.Log("Spawning HUD after delay of 2 seconds.");
        SpawnHUD(spawnObject, 2f);
    }

    IEnumerator RespawnObjectWithDelay(GameObject spawnObject)
    {
        yield return new WaitForSeconds(12);
        Debug.Log("Setting object: " + spawnObject.name + " to active.");
        spawnObject.SetActive(true);
    }
}

```

```

    public void DestroyGameObjectsWithTag(string tag)
    {
        // Find GameObjects with a given tag, and destroy them if found
        var previousConfirmHUDs = GameObject.FindGameObjectsWithTag(tag);
        if (previousConfirmHUDs.Length > 0)
        {
            Debug.Log("Hiding " + previousConfirmHUDs.Length + " GameObjects with tag: '"
+ tag + "'");
            foreach (var previousConfirmHUD in previousConfirmHUDs)
            {
                Debug.Log("Setting object: " + previousConfirmHUD.name + " to
inactive.");
                previousConfirmHUD.SetActive(false);
            }
        }
    }
}

```

ButtonInteraction.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ButtonInteraction : MonoBehaviour {
    private Color standardColor;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    public void Highlight (GameObject a)
    {
        Material objMaterial = a.GetComponent<MeshRenderer>().material;
        if (objMaterial.color != Color.yellow)
        {
            standardColor = objMaterial.color;
        }
        objMaterial.color = Color.yellow;
    }

    public void Unhighlight (GameObject a)
    {
        Material objMaterial = a.GetComponent<MeshRenderer>().material;
        objMaterial.color = standardColor;
    }
}
```

SaveLoad.cs

```
using System.Collections;
using System.Collections.Generic;
using System.Runtime.Serialization.Formatters.Binary;
using System.IO;
using UnityEngine;

public class SaveLoad {

    public static List<Annotation> annotations = new List<Annotation>();

    public static void Init()
    {
        Reset();
    }

    public static void Reset()
    {
        // resets the annotation list to be empty
        annotations.Clear();
        File.Delete(Application.persistentDataPath + "/annotations.gd");
    }

    public static void Save(GameObject b)
    {
        // Transform properties from pushpin (GameObject b) to the Annotation class
        Annotation toAdd = TransformToAnnotation(b);
        Debug.Log("Saving new annotation with id: " + toAdd.id.ToString());
        // Add the new annotation to the list
        annotations.Add(toAdd);

        // Save new list to data file
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Create(Application.persistentDataPath +
"/annotations.gd");
        bf.Serialize(file, SaveLoad.annotations);
        file.Close();
    }

    public static Annotation TransformToAnnotation(GameObject obj)
    {
        // Transform properties from pushpin (GameObject b) to the Annotation class
        Annotation result = new Annotation();
        result.name = "Annotation with ID: " + obj.GetInstanceID().ToString();
        result.id = obj.GetInstanceID();
        result.positionX = obj.transform.position.x;
        result.positionY = obj.transform.position.y;
        result.positionZ = obj.transform.position.z;

        return result;
    }
}
```



```

public static void Load()
{
    Debug.Log(Application.persistentDataPath);
    if (File.Exists(Application.persistentDataPath + "/annotations.gd"))
    {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(Application.persistentDataPath +
"/annotations.gd", FileMode.Open);
        SaveLoad.annotations = (List<Annotation>)bf.Deserialize(file);
        file.Close();
    }
}

public static Annotation CreateFakeAnnotation()
{
    Annotation fakeAnnotation = new Annotation();
    fakeAnnotation.name = "Fake annotation";
    fakeAnnotation.id = annotations.Count + 1;
    fakeAnnotation.positionX = 0.692f;
    fakeAnnotation.positionY = 0.7360002f;
    fakeAnnotation.positionZ = -0.489f;

    return fakeAnnotation;
}
}

```

PushPinController.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PushPinController : MonoBehaviour {
    private GameObject target;

    // Use this for initialization
    void Start () {
        target = GameObject.FindGameObjectWithTag("Shirt");
    }

    // Update is called once per frame
    void Update () {
        // Rotate towards the garment
        Vector3 targetDir = target.transform.position - transform.position;
        Vector3 newDir = Vector3.RotateTowards(transform.right, targetDir, 0.3f *
Time.deltaTime, 120.0f);
        Debug.DrawRay(transform.position, newDir, Color.red);
        transform.rotation = Quaternion.LookRotation(newDir, transform.right);
    }
}

```

Appendix E: Concept designs for tutorial interface



RECORDING #1



0:24 / 0:45

CONTINUE

STOP

REVIEW MODE



Loading the 3D model...

INSTRUCTIONS

Select an area to start reviewing.

Go back to the main screen at all times by pressing the button below the 3D-model.

GOT IT!

NEW RECORDING

When you are ready, press the record button to start recording the verbal feedback.

START RECORDING

RECORDING...



0:24

CANCEL

DONE

RECORDING SAVED

You're recording has been saved successfully.

Duration:

0:45

PLAY RECORDING

DONE

WELCOME

Please choose a mode to continue.

EXPLORE MODE

View given feedback.

START EXPLORING

REVIEW MODE

Provide feedback.

START REVIEWING

References

- [1] PVH Corp., "PVH Corp. (Philips-Van-Heusen)," 2017. [Online]. Available: <http://pvh.com/>.
- [2] P. Corp., "Tommy Hilfiger," 2018. [Online]. Available: <https://www.tommy.com>.
- [3] Y. J. Liu, D. L. Zhang, and M. M. F. Yuen, "A survey on CAD methods in 3D garment design," *Comput. Ind.*, vol. 61, no. 6, pp. 576–593, 2010.
- [4] F. Zhou, H. B. L. Dun, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," *Proc. - 7th IEEE Int. Symp. Mix. Augment. Real.* 2008, ISMAR 2008, pp. 193–202, 2008.
- [5] J. Weft and M. Hill, "The Synthesis of Cloth Objects," *SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 49–54, 1986.
- [6] R. R. Schaller, "Moore's law: past, present and future," *Spectrum, IEEE*, vol. 34, no. 6, pp. 52–59, 1997.
- [7] M. Carignan, Y. Yang, N. M. Thalmann, and D. Thalmann, "Dressing animated synthetic actors with complex deformable clothes," *ACM SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 99–104, 1992.
- [8] B. Lafleur, N. Magnenat-Thalmann, and D. Thalmann, "Cloth animation with self-collision detection," *Proc. IFIP WG 5.10 Work. Conf.*, no. 1986, pp. 179–187, 1991.
- [9] P. Volino, F. Cordier, and N. Magnenat-Thalmann, "From early virtual garment simulation to interactive fashion design," *CAD Comput. Aided Des.*, vol. 37, no. 6, pp. 593–608, 2005.
- [10] I. E. Sutherland, "A head-mounted three dimensional display," *Proc. December 9-11, 1968, fall Jt. Comput. Conf. part I - AFIPS '68 (Fall, part I)*, p. 757, 1968.
- [11] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon, "Windows on the world: 2D windows for 3D augmented reality," *Proc. ACM Symp. User Interface Softw. Technol.*, vol. 93, pp. 145–155, 1993.
- [12] T. P. Caudell and D. W. Mizell, "Augmented reality: an application of heads-up display technology to manual manufacturing processes," *Proc. Twenty-Fifth Hawaii Int. Conf. Syst. Sci.*, pp. 659–669 vol.2, 1992.
- [13] M. B. Spitzer, N. M. Rensing, R. McClelland, and P. Aquilino, "Eyeglass-based systems for wearable computing," *Dig. Pap. First Int. Symp. Wearable Comput.*, pp. 48–51, 1997.

- [14] T. Pfeiffer and P. Renner, "EyeSee3D: A Low-cost Approach for Analyzing Mobile 3D Eye Tracking Data Using Computer Vision and Augmented Reality Technology," *Proc. Symp. Eye Track. Res. Appl.*, pp. 195–202, 2014.
- [15] G. K. Bergey, "Initial treatment of epilepsy: special issues in treating the elderly," *Neurology*, vol. 63, no. 10 Suppl 4, pp. S40–8, Nov. 2004.
- [16] V. Buchmann, S. Violich, M. Billinghamurst, and A. Cockburn, "FingARTips: gesture based direct manipulation in Augmented Reality," *Proc. 2nd Int. Conf. Comput. Graph. Interact. Tech. Australas. South East Asia*, pp. 212–221, 2004.
- [17] K. Lyons, M. Gandy, and T. Starner, "Guided by voices: An audio augmented reality system," *Proc. Int. Conf. Audit. Disp.*, vol. 110, no. July, pp. 57–62, 2000.
- [18] A. Olwal and S. Feiner, "The flexible pointer: An interaction technique for selection in augmented and virtual reality," *Uist'03*, vol. 03, pp. 81–82, 2003.
- [19] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, 1997.
- [20] M. Lee, R. Green, and M. Billinghamurst, "3D natural hand interaction for AR applications," *2008 23rd Int. Conf. Image Vis. Comput. New Zealand, IVCNZ*, pp. 867–872, 2008.
- [21] Dongheng Li, D. Winfield, and D. J. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Work.*, vol. 3, pp. 79–79.
- [22] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the Leap Motion Controller," *Sensors (Switzerland)*, vol. 13, no. 5, pp. 6380–6393, 2013.
- [23] T. Kawashima, K. Imamoto, H. Kato, K. Tachibana, and M. Billinghamurst, "Magic Paddle : A Tangible Augmented Reality Interface for Object Manipulation," *Proc. of ISMR2001*, vol. 195, no. January 2001, pp. 194–195, 2001.
- [24] H. Kato, M. Billinghamurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top AR environment," *Proc. - IEEE ACM Int. Symp. Augment. Reality, ISAR 2000*, pp. 111–119, 2000.
- [25] T. Wright, S. de Ribaupierre, and R. Eagleson, "Design and evaluation of an augmented reality simulator using leap motion," *Healthc. Technol. Lett.*, vol. 4, no. 5, pp. 210–215, 2017.
- [26] A. S. Sultan and M. A. M. Khan, "Feedback in a clinical setting: A way forward to enhance student's learning through constructive feedback," *J. Pak. Med. Assoc.*, vol. 67, no. 7, pp. 1078–1084, 2017.
- [27] H. Hartman, "Scaffolding & cooperative learning," *Hum. Learn. Instr.*, 2002.

- [28] G. Alducin-Quintero, A. Rojo, F. Plata, A. Hernández, and M. Contero, "3D Model Annotation as a Tool for Improving Design Intent Communication: A Case Study on its Impact in the Engineering Change Process," *Vol. 2 32nd Comput. Inf. Eng. Conf. Parts A B*, pp. 349–356, 2012.
- [29] B. Nuernberger, K. C. Lien, T. Hollerer, and M. Turk, "Anchoring 2D gesture annotations in augmented reality," *Proc. - IEEE Virtual Real.*, vol. 2016–July, pp. 247–248, 2016.
- [30] A. Mader and W. Eggink, "A Design Process for Creative Technology," *Proc. 16th Int. Conf. Eng. Prod. Des. Educ. Univ. Twente*, no. September, pp. 1–6, 2014.
- [31] P. Wright, "Human-computer interaction, by A Dix, J Finlay, G Abowd, R Beale, Prentice Hall, 1998," *Softw. Pract. Exp.*, vol. 28, no. 12, pp. 1357–1358, Oct. 1998.
- [32] H. Sharp, A. Finkelstein, and G. Galal, "Stakeholder identification in the requirements engineering process," *Proceedings. Tenth Int. Work. Database Expert Syst. Appl. DEXA 99*, pp. 387–391, 1999.
- [33] T. P. Moran, J. M. Carroll, G. M. Olson, J. S. Olson, and B. Curtis, "Design Rationale: Concepts, Techniques, and Use," *Computers. Cognition, and Work*. L. Erlbaum Associates, p. 488, 1996.
- [34] D. Clegg and R. Barker, *Fast-track : a RAD approach*. Addison-Wesley Pub. Co., 1994.
- [35] Hecla, "Hecla Professional Audio & Video Systems," 2018. [Online]. Available: <https://hecla.nl/>.
- [36] D. A. Bowman, E. Kruijff, J. J. L. J. LaViola, and I. Poupyrev, "An introduction to 3-D user interface design," *Presence Teleoperators Virtual Environ.*, vol. 10, no. 1, pp. 96–108, 2001.
- [37] "Adobe XD CC | UX/UI design, collaboration tool," *Adobe.com*, 2018. [Online]. Available: <https://www.adobe.com/products/xd.html>. [Accessed: 01-Jul-2018].
- [38] "Adobe Photoshop CC | Best photo, image, and design editing software," *Adobe.com*, 2018. [Online]. Available: <https://www.adobe.com/products/photoshop.html>. [Accessed: 01-Jul-2018].
- [39] Gartner, "Top Trends in the Gartner Hype Cycle for Emerging Technologies," *Gartner*, 2017.
- [40] G. A. Lee, C. Nelles, M. Billinghurst, and G. J. Kim, "Immersive authoring of tangible augmented reality applications," *ISMAR 2004 Proc. Third IEEE ACM Int. Symp. Mix. Augment. Real.*, pp. 172–181, 2004.
- [41] M. S. Wagner, "Google Glass : a Preemptive Look At Privacy Concerns," *J. Telecommun. High Technol.*, vol. 11, no. 2, pp. 477–492, 2013.
- [42] George Orwell, "Nineteen-eighty-four," no. 1949, pp. 1949–1950, 1949.

- [43] Lee Mathews, "Chinese Cops Get AR Glasses To Spot Criminals In Crowds - Geek.com," *Geek.com*, 2018. [Online]. Available: <https://www.geek.com/tech/chinese-cops-get-ar-glasses-to-spot-criminals-in-crowds-1730232/>. [Accessed: 14-Jun-2018].
- [44] B. J. Fogg and B. J., "Persuasive technology," *Ubiquity*, vol. 2002, no. December, p. 2, Dec. 2002.
- [45] P. B. Rutledge, "Augmented Reality: Brain-based Persuasion Model."
- [46] M. Neal, J. Cabiria, J. L. Hogg, and S. Pase, "Psychological keys to success in MAR systems," in *2011 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities*, 2011, pp. 1-1.
- [47] Michelle Castillo, "Seattle restaurant bans Google Glass wearers - CBS News," *CBS News*, 2013. [Online]. Available: <https://www.cbsnews.com/news/seattle-restaurant-bans-google-glass-wearers/>. [Accessed: 21-May-2018].
- [48] Richard Gray, "The places where Google Glass is banned - Telegraph," *The Telegraph*, 2013. [Online]. Available: <https://www.telegraph.co.uk/technology/google/10494231/The-places-where-Google-Glass-is-banned.html>. [Accessed: 21-May-2018].
- [49] L. Mitrou, M. Kandias, V. Stavrou, and D. Gritzalis, "Social media profiling: A Panopticon or Omnipticon tool?," *Proc. 6th Conf. Surveill. Stud. Netw.*, 2014.
- [50] Cherlynn Low, "Google's AI advances are equal parts worry and wonder," *Engadget*, 2018. [Online]. Available: <https://www.engadget.com/2018/05/11/google-duplex-ai-tough-questions-answers-now/>. [Accessed: 21-May-2018].
- [51] A. P. Saygin, I. Cicekli, and V. Akman, "Turing Test : 50 Years Later," pp. 463-518, 2001.