Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Virtual and Real-World AOI in AR

Seyda Zarife Öney

Course of Study:	Informatik
Examiner:	Prof. Dr. Sedlmair
Supervisor:	Dr. Kuno Kurzhals, M. Sc. Michael Becher
Commenced:	1. Dezember 2021
Completed:	10. Juni 2022

Abstract

Our work presents a visualization tool, which allows the exploration and comparison of area of interests (AOIs) in an augmented reality (AR) environment. For this purpose, we utilized Microsoft HoloLens 2, which comprises an integrated eye tracker. We extend existing visualization techniques for analysing eye tracking data by taking the AR aspect into account. This requires us to group the environment into real AOIs and virtual AOIs. Virtual AOIs can be identified automatically, however, for real AOIs a manual annotation is necessary. Therefore, our tool provides methods for annotating real-world AOIs efficiently and enables a subsequent analysis of the annotated data, which also facilitates the comparison between multiple participants. To provide a spatio-temporal context, we embedded a combination of two visualization techniques into our tool. The first visualization technique is a timeline visualization, which creates a temporal context. In this timeline, we represent the fixation data by frames depicting the area under observation. Each frame in this visualization gets annotated and then analysed using a scarfplot visualization. For the second visualization technique, we incorporated a gaze replay into our tool. The gaze replay provides the spatial context for the frames, which need to be annotated. We link the timeline visualization with the gaze replay to enable a simple detection and definition of real-world AOIs. To this end, we present a case study, in which we discuss the advantage and disadvantages of our tool. For this purpose, we load gaze data collected from several participants in a pilot study into our visualization tool and analyse it.

Kurzfassung

Wir stellen ein Visualisierungstool vor, das die Exploration und den Vergleich von Area of Interests (AOIs) in einer Augmented Reality (AR) Umgebung ermöglicht. Hierfür verwenden wir die Microsoft HoloLens 2 mit integriertem Eye-Tracker. Unsere Arbeit erweitert bestehende Visualisierungstechniken für die Analyse von Eye-Tracking Daten, unter Berücksichtigung des AR Aspekts. Dies erfordert eine Aufteilung der Umgebung in reale und virtuelle AOIs. Während die Identifikation von virtuellen AOIs automatisch erfolgt, ist eine manuelle Annotation für die realen AOIs notwendig. Unser Visualisierungstool stellt Methoden für eine effiziente Annotation von realen AOIs bereit und ermöglicht eine anschließende Analyse der annotierten Daten, die zusätzlich einen Vergleich zwischen mehreren Probanden erlaubt. Wir kombinieren zwei Visualisierungstechniken in unserem Tool, um einen räumlich-zeitlichen Kontext herzustellen. Die erste Visualisierungstechnik stellt den zeitlichen Kontext her, indem eine Timeline Visualisierung eingesetzt wird. Hier werden Fixationsdaten durch Frames repräsentiert, die den betrachten Bereich darstellen. In dieser Visualisierung werden die einzelnen Frames annotiert und anschließend unter Verwendung einer Scarfplot Visualisierung analysiert. Als zweite Visualisierungstechnik wird eine Gaze Replay Visualisierung eingesetzt, die den räumlichen Kontext für die zu annotierenden Frames herstellen soll. Durch die Verlinkung der beiden Visualisierungsmethoden wird die Erkennung und Definition von realen AOIs vereinfacht. In einer Fallstudie demonstrieren wir unsere Visualisierungstechnik unter Verwendung der Eye-Tracking Daten, die wir aus einer Pilotstudie erhalten, und diskutieren ihre Vor- und Nachteile.

Contents

List of Figures

List of Tables

1 Introduction

Many studies capture the visual attention of participants to gain insights into the perception behaviour and the cognitive processes of humans [HNA+11a]. For this purpose, users' eye movements are recorded with an eye tracker and subsequently evaluated. So far, researchers have focused on analysing gaze data recorded in a real-world environment using mobile eye trackers. Eye tracking technology has developed in recent years to such an extent that even Mixed Reality Head-Mounted Displays (HMD) are equipped with integrated eye trackers. This advancement enables the recording of eye movement data in virtual environments and augmented reality (AR) environments. Our work aims to analyse and compare the visual behaviour of users on real-world objects and virtual objects. Therefore, we used Microsoft HoloLens 2, an optical see-through HMD with an integrated eye tracker that allows users to see the real environment augmented with virtual objects.

Depending on the research question, different methodologies for analysing the gaze data have been introduced. However, one of the main approaches are the quantitative analysis using statistical measures and the qualitative analysis using visualizations [BKR+17]. The complexity of the data generated by eye trackers has been increasing over the years due to the advancement in eye tracking technology. This increasing complexity often causes statistical measures to be insufficient to interpret the data adequately [Bur16]. In particular, mobile eye trackers that provide additional spatial context can be poorly analysed by statistical means. In order to analyse patterns in gaze data and construct hypothesis, the visualization of eye movement data is essential [KBPW15].

For visualizing gaze data, we can distinguish between two methods, namely point-based methods and Area of interest (AOI)-based methods. The former method enables the analysis of the overall gaze distribution, while the latter enables semantic analysis by defining AOIs onto the stimulus and allows the investigation of AOI-based metrics, which provide additional information about transitions and relations between AOIs [BKR+17]. Per our knowledge, previous work considered AOI-based visualization techniques for 2D stimuli in the real environment([KHW14], [KHH+15a]) and neglected the analysis of gaze behaviour on AR environments. Therefore, we developed an AOI-based visualization approach, which enables the analyses of visual attention on the real-world AOIs and virtual AOIs providing a 3D context. The main contributions of our work are as follows:

- Comparison between real-world AOIs and virtual AOIs
- Visualization with 3D scene representation of stimulus
- AOI timeline visualization linked with 3D gaze replay

Structure

This thesis is structured as follows:

Chapter 2 - Foundations: Here, we give the fundamentals necessary for understanding our work. For that purpose, we first introduce augmented reality, an then address eye-tracking-specific aspects. These include well-known AOI-based visualization approaches as well as common eye-tracking metrics.

Chapter 3 - Related Work: describes work that has used different approaches to map 3D gaze data to the physical environment. It also presents previous work on visualization techniques developed using AOI or image-based approaches that considers similar aspects to our work.

Chapter 4 - Task and Concept: The purpose of our work is described in detail here, as well as a concept for realizing the visualization technique.

Chapter 5 - Data Processing: This chapter describes the hardware and software components used to collect data in the pilot study.

Chapter 6 - Visualization Design: Here, we discuss the individual components of our visualization tool in depth, as well as possible interaction methods for annotation and analysis of the AOIs.

Chapter 7 - Case Study: This chapter first describes the pilot study's structure and procedures. After that, the pilot study data will be considered and analyzed in a case study. The purpose of the case study is to evaluate the usability of our visualization technique.

Chapter 8 - Conclusion and Outlook: outlines our research topic and offers directions for future work.

2 Foundations

This chapter introduces the topics necessary for understanding our work described in the upcoming chapters. The first part describes the basic concept of augmented reality and introduces well-known augmented reality technologies. The second part of this chapter contains the main part of our work. Here, we first discuss eye-trackers in general and then introduce well-known eye-tracking metrics. Finally, important visualization concepts are introduced.

2.1 Augmented Reality

According to Azuma's definition, augmented reality (AR) is the enrichment of the real world by computer-generated additional objects, the so-called virtual objects [Azu97]. In this context, an AR technology should fulfill the following three requirements:

1. combines real and virtual objects
2. runs interactively in real time
3. registers virtual content in/with the real world

A well-known technology for the use of AR is the see-through Head Mounted Display (HMD), developed by Sutherland in the 1960s [Sut68]. The use of an HMD provides an immersive environment by superimposing virtual elements onto the user's view of the environment. Optical see-through HMDs such as the Microsoft HoloLens 2 can be used in a variety of application areas due to their multiple functions.

2.2 Eye-Tracking

Eye trackers are used to record gaze data from users. The recorded gaze data contains different measurements, which can be utilized to analyse it. Depending on the measure, necessary metrics can be derived, which are detailed in the following. A detailed overview of eye tracking measures and their corresponding metrics can be found in [HNA+11a].

2.2.1 Eye-Tracking Metrics

Fixations A fixation can be described as the aggregation of gaze data [BKR+17], where the eye is relatively stationary [PB06]. According to [HNA+11a] the timespan of a fixation is between 200-300ms. During this period, information on the stimulus gets processed. Known metrics for fixations are the number of fixations, fixation duration, and fixation position [BKR+17].

Saccades A saccade is described as the distance between two fixations. It is a rapid eye movement, which can last about 30-80 ms. During a saccade, there is no encoding of information. Common metrics for saccades are the number of saccades, the saccadic amplitude, duration of saccades and the velocity of saccades. [BKR+17]

Scanpaths A scan path can be described as a saccade-fixation-saccade sequence [PB06]. Based on this sequence, the search behaviour of users can be analysed. Typical metrics for scanpaths are the area covered by a scanpath (convexhull), the scanpath length and the scanpath duration. [BKR+17]

2.2.2 Area of Interest

A detailed analysis of gaze data requires the definition of AOIs. These specify parts in a stimulus, which are of special interest [DLS+14]. In an AOI-based visualization approach, only the eye movements on the AOIs are visualized and analysed, while the rest gets neglected [PB06]. AOIs can be defined based on the semantic information a stimulus provides before or after an experiment. Common metrics used for AOIs include the number of transitions between AOIs, the dwell time within an AOI, and the AOI hit. Here, transitions describe the saccadic movement from one AOI to another and the AOI hit describes if a fixation is on an AOI or not. [BKR+17]

2.2.3 Visualization Techniques

We distinguish between AOI-based and point-based visualization techniques, as mentioned in the introduction. Since our work introduces an AOI-based visualization technique, we will discuss some existing AOI-based visualization techniques in here. These methods make use of data that has previously been AOI-labeled.

Scarf plot A scarf plot is based on a timeline visualization with time on one axis and participants on the other [HNA+11b; KHW14; RD05; RHOL13b]. It represents the duration of fixation to the AOIs. By creating a color map for the AOIs, the time intervals in the scarf plot are colored accordingly, allowing the sequence and length of the AOIs to be analyzed. Because multiple participants are considered, they can be compared. The drawback of the scarf plot is that the number of AOIs that may be evaluated for a meaningful comparison is limited.

There are several types of scarf plots. While scarf plots are used in [KHW14] to depict the scan paths of multiple individual participants, AOIs are represented by thumbnail images in [TTS10a] and participants are considered in groups. [BKR+17]

AOI Timeline In comparison to scarf plots, an AOI timeline is built on a timeline visualization with the second axis represented by AOIs. Individual participants' data can be visualized, or the data of multiple participants can be aggregated. Work can be found where the time periods are colored [CN00; HNA+11b; Hol01; WFE+12]. This kind of visualization can be used to compare scan paths. In this case, the number of participants considered is limited to the stated number. [BKR+17]

Transition Matrix A transition matrix [GK99] is a relational visualization technique for examining the transitions between AOIs. The number of transitions between all AOIs is presented in pairs in the matrix. The distribution of the number of transitions in the matrix can be used to analyze the searching behavior of individual participants. Search patterns can be detected using color mapping of the number of transitions [BKR+16; KSK13].

When comparing multiple participants, their matrices can be concatenated, enabling visual comparison [GH10]. [BKR+17]

Tree Visualization A tree visualization can be applied as a relational visualization technique to examine the AOI sequences of individual participants. The nodes of the tree visualization represent the AOIs that are connected according to their transition. There are several variants of tree visualization. Besides typical transition trees [RHOL13b; TTS10b; WHRK06], there are also transition trees where the AOIs are represented by thumbnails [RHOL13c]. [BKR+17]

2.3 3D Scene Reconstruction

For a 3D scene reconstruction, a point cloud of the scene needs to be generated. This can be done using hardware like a laser scanner or by taking images of the scene and applying computer vision (CV) methods onto the images. The laser scanner is capable of providing a point cloud directly [ML18]. Depending on which camera was used and whether a single image or multiple images were taken, the image-based 3D reconstruction approaches can differ. In [HWH19] algorithms from literature for computer vision based 3D reconstruction are presented. They differentiate between single still image based approaches, RGB-depth image approaches, multiperspective of 2D images approaches and approaches for video sequences. Based on the input, the algorithm for reconstructing a 3D model changes slightly. However, the general process includes the image acquisition, the image preprocessing by applying Structure from Motion (SfM) methods like camera motion estimation, feature extraction, feature matching and more onto the images [ML18]. In the next step, a 3D point cloud is obtained and textures and meshes can be generated to get a 3D model [HWH19].

3 Related Work

In the previous two chapters, we have provided an introduction to our work and clarified the necessary foundations for it. In this chapter, we present related works from two areas that are relevant to our work. The first part presents works which have used different approaches to create 3D scene reconstructions and to map the gaze within the scene. The second part of this chapter consists of works on image-based and AOI-based visualization approaches. They are compared with our visual approach.

3.1 Gaze Mapping based on 3D Reconstruction

Mapping the gaze into 3D space plays an important role in the spatio-temporal analysis of 3D gaze data. In the following, we present works that differ in their approaches regarding the modeling of the viewed 3D environment as well as the localization of the camera.

Some works, like ours, consider many viewers in the same static scene. Parikh et al.[PLX+19] proposed one such approach where the 3D scene was reconstructed using Structure of Motion (SfM). The camera localization approach consisted of localizing the images used for reconstruction. Nearest neighbor search and bundle adjustment were used to localize an image. They developed a convolutional neural network to track eye movements and estimate the gaze vector. The projection of the 2D gaze onto 3D space was used to localize the gaze in 3D space. This approach provided accurate localization of gaze from multiple viewers.

On the other hand, Pieszala et al.[PDP+16] developed a pipeline for camera localization that combines global (LIDAR scanning) and local (dense SLAM-based) registration techniques. In this method, local alignment of camera frames was first performed using the dense SLAM-based alignment algorithm. For global alignment, LIDAR scans were performed using a FARO Focus3D laser scanner. Then, the camera images were aligned with the LIDAR 3D data. The pipeline provides the capability for manual labeling of the regions of interest (ROI). It also allows for further refinement of the alignment. This approach provides an accurate estimation of the camera position and yields a dense 3D model. The resulting visualization system includes a replay with an ROI visualization similar to ours with additional gaze statistics information.

Yan and Tamke[YT21] did not consider multiple viewers in their work. However, similar to us, they scanned the environment twice - with the HoloLens2 and the LaserScanner Faro. The laser scanner created a point cloud. This was voxelized and aligned with the mesh model generated by the HoloLens. Evenly spaced colored points were overlaid over this model to create a heat map.

The approach of Hagihara et al.[HTA+18] differs from the other approaches described previously. They present a method for objectwise gaze mapping, where the gaze distribution is visualized on the real objects. The 3D objects are automatically segmented in advance using a SLAM-based method, then dense 3D surfaces are created. Based on Visual SFM, the 3D feature map is reconstructed, which enables the estimation of the camera position.

If a static environment contains dynamic objects, it is considered a dynamic environment. Attention analysis in a dynamic environment often requires the additional use of CV techniques. This can be observed in the work of Oishi et al.[OKYB21]. They have developed the 4D attention framework. At first, they created the 3D model of the environment using the LaserScanner Focus3D. For the localization of the camera, its position in the 3D map was tracked using the C* method [R1] in combination with the IMU measurements. The ID texture mapping was applied for a uniform and efficient gaze projection. Here, the texture has a unique ID in each of its pixels. The gaze on dynamic objects is determined by rendering these objects into the 3D environment with a given mesh model and ID texture, their position is determined by applying CV methods (SIFT). Evaluation of this approach revealed high accuracy in attention mapping.

Another work considering dynamic environments was proposed by Kraus et al.[KPM+20]. They built the framework with the goal of facilitating the work of police officers in inspecting digital data from a crime scene. They combined all video and imagery to create a 3D reconstruction. First, they created a sparse reconstruction using COLMAP, a Structure of Motion (SfM) pipeline. Then, a textured surface model of the scene was created using the Multi-View Stereo (MVS) pipeline [ope]. If geospatial data was not available, the reconstructed model could be manually scaled and aligned. In our work, we also manually aligned the position and scale of our 3d model.
In order to segment the dynamic objects in the model, an orthogonal depth estimation approach was chosen in combination with MaskRCNN. A neural network-based approach, OpenPose, was used specifically for human detection. It was used for detecting people who were partially occluded. The framework also includes a tool for manual annotation. Similar to our work, a plane can be manually placed, scaled and moved accordingly.

Scene reconstruction also plays an important role in the development of remote collaboration systems. Bai's work[BSYB20] considered this, where a local worker shares his physical workspace with a remote expert and his hand and gaze positions are visualized. The local worker's environment is presented to the expert as a real-time 3D panorama. A sensor cluster of 8 RGB depth cameras is formed for this purpose. The point clouds of the sensors are merged into a hemisphere shape to reconstruct the scene. The panoramic scene is mapped onto a VR headset used by the expert. The coordinate systems of the two users are equally aligned using an image tracking calibration method.

3.2 AOI-based and Image-based Visualization Techniques

In recent years, numerous visualization techniques have been developed to perform spatio-temporal analysis and to detect and define AOIs. Static visualizations have been used that should facilitate a fast AOI identification.

Kurzhals et al.[KHH+15a] presented an image-based approach for static and dynamic stimuli that does not require explicit definition of AOIs. The approach collects image data around all gaze points and orders them sequentially on a horizontal timeline. The visualization of the timeline is extended with additional views to provide a global context. For this purpose, the timeline is linked to a video player with bee swarm visualization. Clicking on gaze stripes highlights the respective regions in the video player. Stimulus context is available through the thumbnails, which enables the detection of outliers and patterns. In addition, this approach provides a general overview of the scene. Unlike conventional techniques , there is no visual overlap. Multiple viewers can be considered in the analysis. This approach shows good scalability in terms of number of viewers and length of gaze sequences.
Similar to this work, we adopted an image-based method and linked to gaze replay to provide a global context. The difference is that the Gaze Replay consists of a 3D reconstruction of the scene, rather than a video, in which the gaze behavior of multiple viewers can be considered. Gaze Stripes is classified as a point-based visualization according to Blaschek et al.[BKR+17], while our work is AOI-based. In our work, the detection and definition of AOIs is required for the analysis of gaze behavior. Based on the gaze stripes approach, Kurzhals et al.[KHH+15b] proposed a fixation image chart for the analysis of static and dynamic stimuli. A more compact timeline is used here by aggregating gaze data to fixations and mapping one thumbnail image per fixation. Since temporal synchronization is lost by aggregation, time streams were used. Based on this, we also represented each fixation by a thumbnail in our visualization approach. To preserve the temporal synchronization, each fixation is surrounded by a rectangular shape. The length of the rectangle represents their fixation duration.
Further, the fixation image chart is extended by providing interactive filtering capabilities and descriptive statistics of fixation metrics. While this approach does not require an explicit definition of AOIs, they instead considered labels that are manually defined for each fixation. This allows the detection of fixation patterns. Pontillo et al. [PKP10] have previously assigned labels to fixations for semi-automatic classification of fixated regions, similar to this approach. The goal was a semantics-based encoding of fixations. During database training, a set of fixations were manually encoded and objects were defined that contained the image regions considered as well as its color histograms. Using a color histogram intersection method, the remaining fixed regions were matched with the defined objects and the object with the largest matching value was assigned. The labeled data could be exported and used in further analyses.

A well-known visualization technique for a spatio-temporal analysis is the Space Time Cube visualization [BKR+17]. The static visualization allows to find time sequences faster and to detect possible AOIs faster. Kurzhals and Weißkopf showed this in their work[KW13]. In their visualization approach with multiple coordinated views, STC visualization is used in combination with computer vision algorithms with optic flow estimation. CV technique is applied to find Shot Boundaries automatically. These are used in combination with spatio-temporal clustering to identify AOIs. Their visualization technique is intended for the analysis of dynamic stimuli with multiple viewers.

A similar visualization approach combined with AOI-based visualization is ISeeCube [KHW14]. It enables the spatio-temporal analysis of dynamic AOIs by using a space-time cube visualization with linked timeline visualization. Here, AOIs are defined by placing bounding boxes in the video at different key positions, and the bounding box is interpolated between the key positions. For advanced analysis, it shows how long an AOI was visible before the viewers looked at it. In addition, a scarfplot for each viewer is integrated into the timeline visualization, which should facilitate a scanpath analysis. We also create a scarfplot-like timeline for each viewer in our approach: if each fixation has been annotated, the rectangle around the thumbnail image can be colored according to the AOI. Thus, our visualization provides a scarfplot view with local context.

EyeC [RHOL13a] is a visualization system that was used for the analysis of psychological experiments. The important aspect here is that a comparison between viewer groups is possible in the analysis. Thus, to compare fixation duration of AOIs between groups, the data can be aggregated and bar graphs can be created. This approach can be applied to our work with the difference that the virtual and real AOIs are considered as groups. In our work, the individual AOIs are categorized into virtual and real AOIs, and thus can be aggregated.

4 Task and Concept

4.1 Task

We attempt to develop a visualization technique based on AOIs to examine visual attention in augmented reality scenarios. A distinction between real and virtual AOIs is necessary here due to the AR aspect. Since the gaze rays directly hit the virtual objects and identify them, virtual AOIs can be identified automatically. Manual annotation is required for the identification of real AOIs. Therefore, it is part of our work to integrate annotation support into our visualization approach.

The task was divided into two parts. The first part was to collect eye-tracking data that would later be used for visualization. Thus, the goal was to create an AR scenario for the pilot study in which the tasks support the viewing of real and virtual objects. This part is described in more detail in Chapter 5. Microsoft HoloLens2 with an integrated eye tracker was employed as the AR technology.

The acquired data should be used in the second phase of our work to build an AOI-based visualization technique that provides spatio-temporal context and allows comparison of multiple viewers. The visualization technique is explained in detail in Chapter 6.

In the next section, a concept for the realization of the visualization technique is described in more detail.

4.2 Concept

The concept for creating the visualization technique can be explained using the visualization pipeline illustrated in Figure 4.1. The pipeline describes how to perform a data transformation after data collection to extract and filter the necessary data, and then map it to visual structures in the visual mappings step. In the view mappings step, different views can then be provided on the visualized data.

In the following, we apply the transformation stages to our visualization concept:

Data Transformations First, data is collected in a pilot study. A spatial mesh of the scene environment is saved in this step. During the study, the eye-tracking data is logged and a video recording is captured with the HoloLens front camera. In the data transformation, fixations are determined from the eye-tracking data and stored. Thumbnail images of the fixation areas are extracted from the video recording.

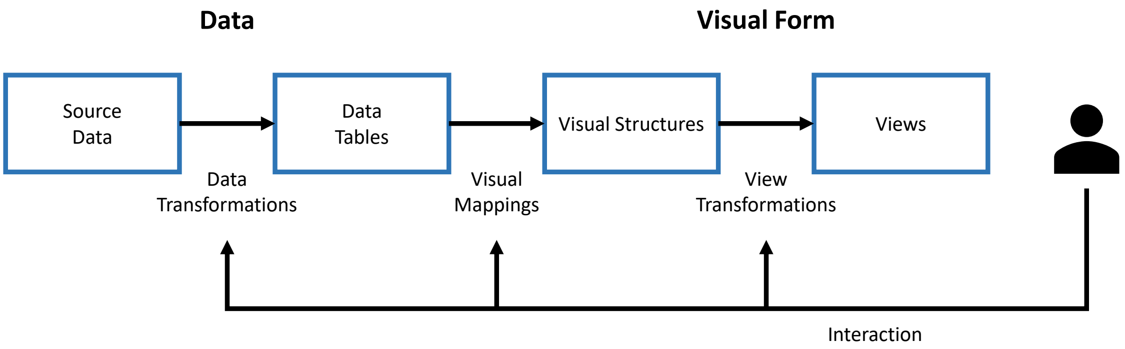


Figure 4.1: Visualization pipeline as proposed by Card et al [Car99]

Visual Mapping In this step, we use the extracted data to create a visualization. Since we intend to create an AOI-based visualization technique with spatio-temporal context, we combine two visualization approaches. First, we choose a timeline visualization, namely scarf plot visualization, a common approach for analyzing AOI-based data. The second visualization approach is gaze replay visualization. The two approaches must be connected to consider spatio-temporal aspects.

Figure 4.2 shows a rough draft of the visualization we create. The fixations, represented by thumbnails or frames in the scarf plot, are positioned sequentially based on their timestamp. In the gaze replay, the 3D scene is visualized using the spatial mesh to provide a spatial context for the fixations. By applying interaction techniques such as filtering, selection, and navigation, the user can identify and define AOIs to annotate in the timeline visualization.

For the analysis of scarf plots, it is necessary to encode fixations with visual variables. Fixations are mapped to AOIs, and AOIs to colors. Since we distinguish between virtual and real AOIs, they can be colored differently.

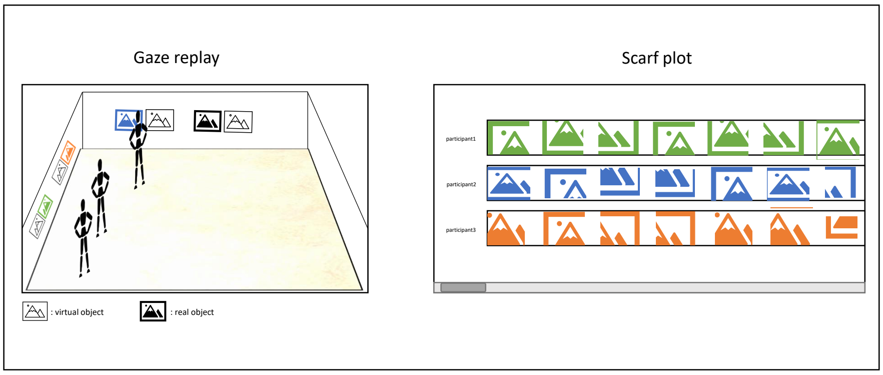


Figure 4.2: Sketch of the desired visualization

View Transformations Multiple views in the gaze replay are required if the user wants to observe the AOIs in the 3D scene. It is also possible to switch between views in the timeline visualization by using interaction techniques such as zooming, filtering, and navigation for the analysis.

The realization of the described visualization process, from data preparation to visualization design, is covered in the following chapters (Chapter 5 and Chapter 6).

5 Data Processing

As described in chapter 4, we must first collect eye-tracking data before we can create our visualization and apply the data to it. This chapter will describe the data collection methods employed. The hardware and software components, as well as the tools utilized, are discussed.

5.1 Hardware and Software Components

To carry out the pilot study, a mobile AR technology that supports eye-tracking and generates 3D gaze data is required. The Microsoft HoloLens 2 with built-in eye-tracker was employed in our pilot study. The AR scene is developed in Unity and then deployed on the HoloLens 2. Three toolkits are required for the AR application to be implemented. All of the mentioned components are described below.

5.1.1 Microsoft HoloLens 2

Microsoft HoloLens 2[1] (see Figure 5.1) is a mixed reality headset with built-in computing capabilities. It comes with a Qualcomm SnapDragon 850 SoC and a second-generation holographic processing unit (HPU 2.0) that supports real-time computer vision. All computer vision algorithms used by HoloLens, like head tracking, hand tracking, gaze tracking, and spatial mapping, are executed on the HPU [UBG+20].

Figure 5.1: The Microsoft HoloLens 2 device. [RKT22]

[1] https://www.microsoft.com/de-de/hololens last seen: 09.06.22

The HoloLens has several sensors built-in, including an RGB camera with 8 MP and 1080p30 for image and video recording. HoloLens 2 includes grayscale visible light cameras (VLC), which are used for head tracking and the real-time visual-inertial SLAM algorithm [UBG+20].
The tracking system employs two of these cameras to detect visual features and triangulate based on the features. The triangulated position and IMU data can be utilized to determine the position of the HMD in the environment. [**51**] The additional two cameras are used to extend the field of view for feature tracking [UBG+20]. Two infrared cameras are employed for eye-tracking and iris recognition. The HoloLens also includes a depth camera with a 1-megapixel time-of-flight sensor for detecting the environment and hand tracking. The frame rate of the camera varies depending on the depth sensing. A frame rate of 45 FPS is available for near-depth sensing and a low frame rate of 1 to 5 FPS for far-depth sensing, which is used for spatial mapping. [UBG+20]

The Inertial Measurement Unit (IMU) is another sensor that consists of an accelerometer, a gyroscope, and a magnetometer:

- accelerometer: contains acceleration along the axes
- gyroscope: contains angular rotation around the axes
- magnetometer: contains the absolute orientation estimation along the axes

HoloLens offers a Research Mode for computer vision and robotics research. When the HoloLens's Research Mode is enabled, the measured values of the IMU, as well as other sensor streams, are accessible.

The mixed reality headset can be controlled via speech and gesture. Hand tracking detects all hand movements, allowing a virtual element to be selected simply by clicking on it. Drag and drop can be used to move objects. To access the menu, a virtual Windows icon appears on the wrist, which must be clicked.

5.1.2 Unity3D

Unity3D [2] is a cross-platform game engine created by Unity Technologies. It's used to make 2D and 3D games, as well as AR and VR apps.

To develop an application, a Unity project is first created, which later contains all the elements of the application. Scripts, 3D models, texture materials, shaders, and other assets are included. A project can include one or more scenes. Depending on its complexity, an application can be developed in a single scene or in multiple scenes that are connected together. [NK19]

When a scene is created, it already contains GameObjects that represent the camera and lighting. GameObjects are used to store players, 3D environments, object managers, UI elements, and other elements in Unity. To give a GameObject individual properties, several components must be attached to it. Every GameObject has a Transform component that controls its position, rotation, and scaling in Unity Space. The MeshRenderer is used to define the materials for an object. The

[2] `https://unity.com/de` last seen: 09.06.22

Collider is one component that is very important for identifying virtual AOIs. A gaze ray is sent from the eye to the object being observed, and it is hit if the object has a Collider. Objects that have been hit can be identified.[NK19]

To integrate AR into the Unity application, additional components are required. As a result, the Mixed Reality Toolkit is utilized. It is presented below.

Mixed Reality Toolkit (MRTK)

Microsoft developed the Mixed Reality Toolkit[3] (MRTK), which is used to build mixed reality applications in Unity projects. It can be imported manually via the GitHub website or automatically using the Mixed Reality Feature Tool. In our pilot study, we manually loaded the MRTK 2.7.2.0 into our project.

The toolkit includes various features that can be added to the HoloLens application. The MRTK has an Input System feature that detects hands or speech as input sources for AR interaction. Other important features include eye-tracking, UI controls, hand tracking, and spatial awareness.

When using MRTK in Unity, several MRKT-specific GameObjects with mixed reality functionality are loaded into the scene. Therefore, the parent GameObject of the camera is the *MixedReality-Playspace* GameObject. This GameObject is used by other MRTK components to perform world and local space transformations. [Micc]

The *MixedReality Toolkit* GameObject is in charge of the lifetime and runtime of its services, which includes, for example, the sequence in which its services are initialized. The services can be configured using MRTK's predefined profiles. For HoloLens 2, MRTK provides a default profile [Mice]. Adjustments are also possible. For the pilot study, we used the *WLT HL2 MRTKProfile* provided by World Locking Tools (see Section 5.1.3) and customized the following configuration profiles:

- In the Experience Settings, the *Target Experience Scale* was changed from Room to World to allow the HoloLens to travel more than 5 meters. When the application starts, this setting sets the coordinate system's origin to the HoloLens position. [Mica]
- In Spatial Awareness, the *Observation Extents* for the *Windows Mixed Reality Spatial Mesh Observer* were increased to *X:10, Y:10, Z:10* to ensure that the mesh in our pilot study space was still recognized (see Figure 5.2). Second, the Level of Detail was modified from *Coarse* to *Fine* to get the most accurate spatial mapping.
- For the Mixed Reality Camera Settings, *Render from PV Camera* was turned on. The toolkit ARETT (see Section 5.1.4) needs this feature to display the hit gaze as a green sphere in the video recording while it is hidden from the user.

[3] https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05 last seen: 09.06.22

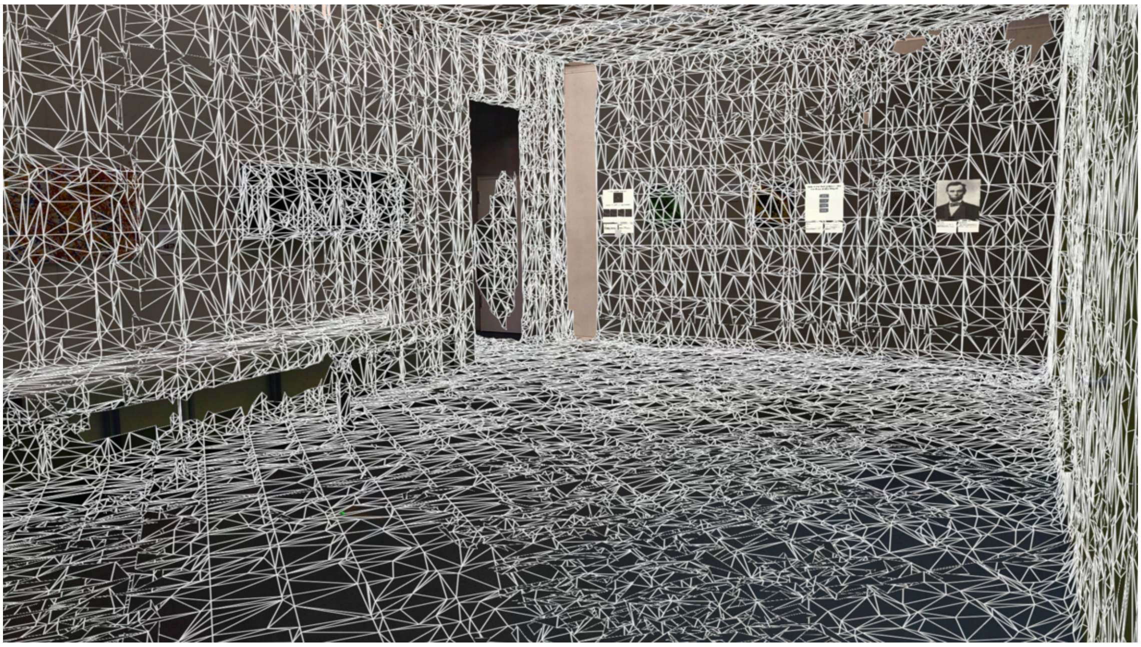


Figure 5.2: The virtual scene with visible spatial mesh of the room is shown. The mesh is visualized using the Spatial Awareness feature of the MRTK.

In Unity, a pipeline must be chosen before the HoloLens application can be built. There are two pipelines: legacyXR and XR SDK. Unity versions after 2020 will only use the XR SDK, whereas Unity versions before to 2019.3 will continue to use the legacyXR. All versions in between can use either of the two pipelines. In our pilot study, we used Unity 2019.4.34f and chose the legacyXR pipeline because we had to use the ARETT toolkit, which only supports legacyXR.

5.1.3 World Locking Tools (WLT)

The HoloLens is equipped with sensors that enable head-tracking and environment understanding. The environment is continuously tracked, and our understanding of it is updated as we move through space. When virtual objects are placed in real space, updating the understanding of the space might lead the objects to drift in the physical environment over time. However, the relative positions between virtual objects stay unchanged. Spatial anchors can be used to preserve the positions of virtual objects in physical space by adjusting their coordinates. [Micg]

WLT[4], which is a standalone feature of MRTK, places various spatial anchors throughout the virtual scene rather than putting spatial anchors for individual virtual objects. When moving through space, the head coordinates are adjusted to the updated environment instead of the object coordinates altering. [Micg] The main difference between spatial anchors and WLT is that spatial anchors move virtual objects in Unity Space to keep their position in physical space. WLT locks the entire virtual scene to the physical environment. [Mich]

[4] https://docs.microsoft.com/en-us/mixed-reality/world-locking-tools/documentation/overview last seen: 09.06.22

Another issue with HoloLens is the scale problem. It shows how, when a user walks from point A to point B, the tracked distance does not match the real distance walked in physical space. For the HoloLens, the error boundary is -+ 10%. I. e., if the actual distance between points A and B is 10 meters, the HoloLens may interpret it as being between 9 and 11 meters. [Micf] To solve this problem, WLT provides the Space Pinning feature. The space pin moves the entire Unity space to align a virtual object to a specified physical visual feature. Using multiple space pins can help to solve the scale problem. In addition, smooth interpolation is performed between space pins. The scale error is reduced by taking these steps.

In our pilot study, we specifically employed WLT's QR Space Pin Sample project [Micd]. QR markers are used in this project as physical visual features to align virtual objects with. First, we set four physical QR markers around the room (see Figure 5.3). We defined a starting point in the center of the room and measured the positions of the QR markers in relation to it. We created virtual markers in the virtual scene and assigned measured positions to them. In this scene, we built the virtual elements for the pilot study by placing the virtual objects relative to the virtual markers.

When the virtual scene application is launched, the Unity space's origin is defined as the head position. When the user scans the physical QR markers with the HoloLens one after the other, the coordinate space is transformed to align the virtual markers with the recognized physical markers. This moves the coordinate space's origin from the head position defined at the start of the application to the physical start position (see Figure 5.4). As a result, the coordinate space is defined the same for each participant.

WLT's QR Space Pin Sample project was integrated to meet the following pilot study requirements:

1. The virtual objects must be placed as correctly as possible relative to the physical environment so that the AR scene looks the same for all participants.
2. A common coordinate system for all participants is required so that all participants can be simulated later in the Gaze Replay.

We employed another WLT feature to support the pilot study. WLT offers persistence. Spatial anchors are saved locally in the HoloLens' anchor store and can be reloaded from there into the application over and over again to offer persistence across sessions, ensuring that virtual objects keep their position when the application is reloaded. In particular, in the QR Space Pin Sample project, the world-locked scene may be saved in the HoloLens file explorer as a *frozenWorldState* file. As a result, it was sufficient to align the virtual scene with the physical space once using the physical QR markers and then create a *frozenWorldState* file. This file was loaded for the participants during the study to ensure that the virtual scene was accurately aligned. On the one hand, this saved time during the study, while on the other, it prevented a possible deviation during the recognition of the QR markers. This is because the relative position of the QR markers to each other may differ on each run.

5.1.4 Augmented Reality Eye Tracking Toolkit for HMD (ARETT)

ARETT [KBM+21] is an open source toolkit implemented in Unity that can be imported as a package into a Unity project. It is suitable for eye-tracking research in AR environments and has been implemented specifically for the HoloLens 2. However, it can also be used on other devices by making adjustments in the program.

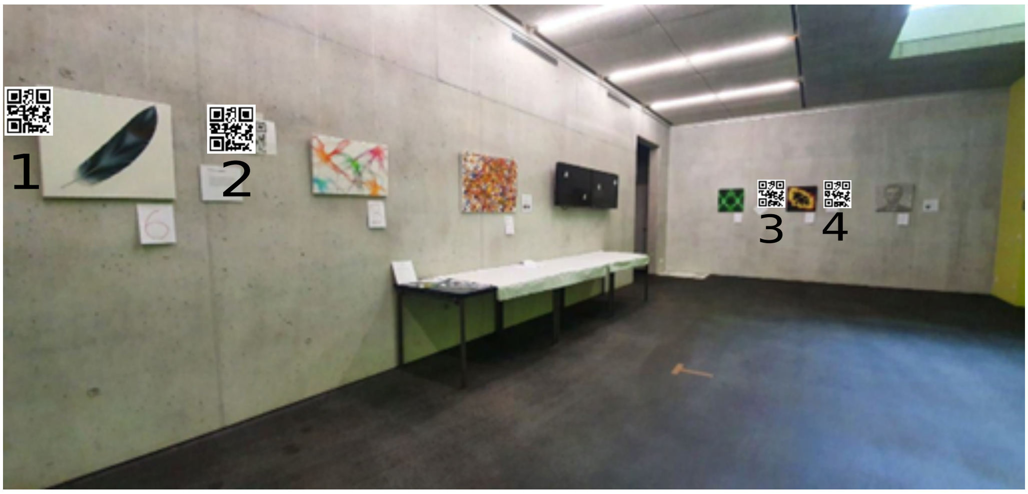


Figure 5.3: There are four qr markers in the room. Two of them surround the image in the front and two in the back. They are viewed in numerical order.

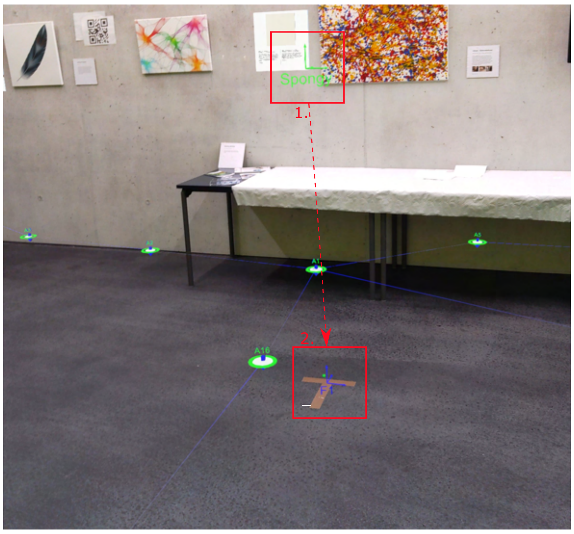


Figure 5.4: The origin(1.), which is defined by the head position at the start of the application, is transformed to the floor (2.) by WLT. The desired origin position is represented by the T mark on the floor.

MRTK provides eye-tracking data for the HoloLens, but it is better suited for gaze-based interaction. The problem is that the sampling rate is tied to Unity's update rate, so there is no guarantee of a fixed sampling rate and gaze data can be lost.

Because ARETT utilizes the native eye-tracking API for UWP, it can achieve stable data rates, unsmoothed data, and better control. The timestamp has a precision of 100 nanoseconds per data point. The HoloLens has a sampling rate of 30 Hz, corresponding to a sampling interval of 33.33 ms [Micb]. The spatial accuracy of the predicted gaze is about 1.5 degrees [Micb].

ARETT's architecture consists of multiple components. A schematic depicting the components' interdependence is presented in Figure 5.5. The data access layer is the first component. It queries new gaze samples from the UWP API every 10 milliseconds to provide real-time eye-tracking data. A gaze sample is obtained at a sampling interval of 33.33 ms due to the sampling rate of 30 Hz. The data provider component receives the gaze samples and processes them further.

The gaze samples include the gaze origin, gaze direction, and a timestamp. This data is expanded in the data provider layer by mapping the gaze sample to a gaze ray in the HoloLens coordinate system. A ray is sent from the gaze origin to the gaze direction in this step, and virtual objects with colliders are hit. This captures information on the hit objects, such as the object's name, global and local hit points, position, and other data. ARETT also allows you to define AOI colliders in the Unity scene. We did not use this functionality of ARETT because the virtual objects in our scene already had a collider.

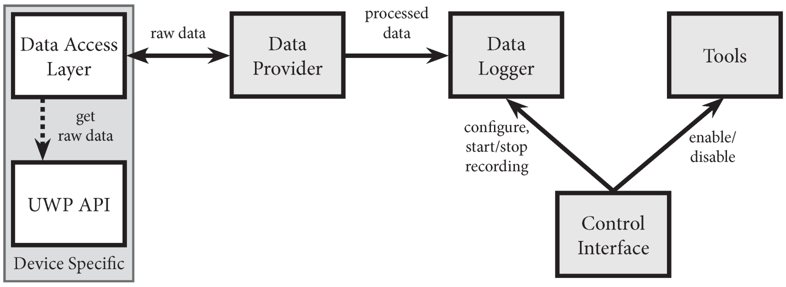
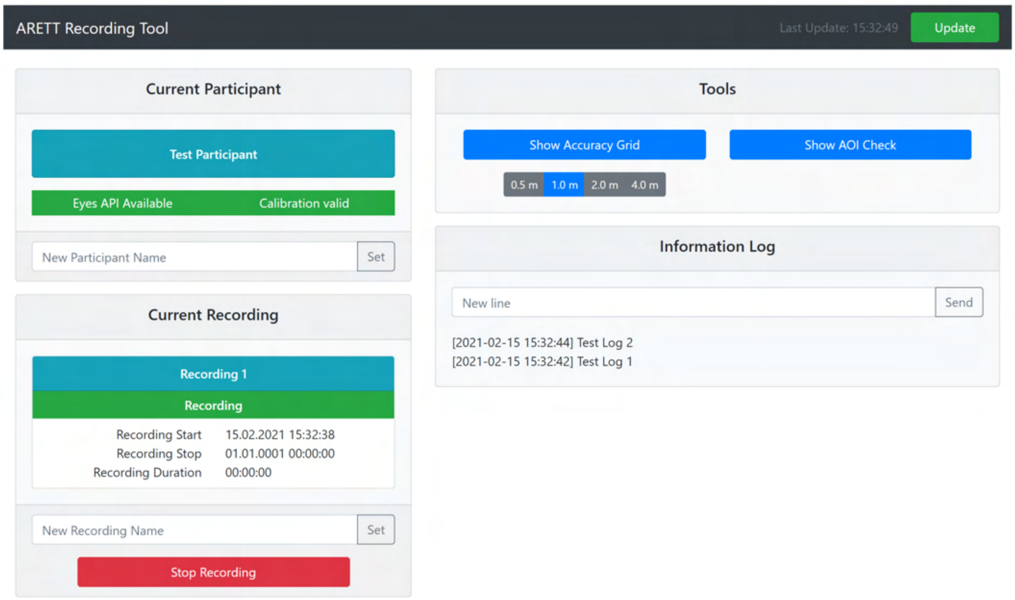


Figure 5.5: The diagram represents the relations between the main components of ARETT. [KBM+21]



Recording Start 15.02.2021 15:32:38
Recording Stop 01.01.0001 00:00:00
Recording Duration 00:00:00

[2021-02-15 15:32:44] Test Log 2
[2021-02-15 15:32:42] Test Log 1

Figure 5.6: The control interface manages the recording's start and stop. [KBM+21]

In the Data Logger component, samples are subscribed and stored in a CSV file for each participant. The metadata for the recording is saved in a JSON file.

ARETT provides a web-based control interface (see Figure 5.6) that is accessible over a local network. The interface is used to name the CSV file and to start and stop the recording. It is also used to determine whether the eye calibration is valid; otherwise, the interface is used to trigger the eye calibration process. Further functions can be derived from [KBM+21].

In addition, ARETT has a feature that allows you to see the gaze point in a video recording. A virtual 2D camera is constructed to imitate the HoloLens' front-facing camera for this purpose, and the 3D gaze point is projected onto it. The gaze point is presented as a green sphere in the video recording (see Figure 5.7), but it is not visible to the participant during the recording. To use this feature, as described in Section 5.1.2, the render from PV Camera must be enabled. For each gaze sample, the position of the projected gaze point on the virtual camera is saved in the CSV file under the name gazePointWebcam.

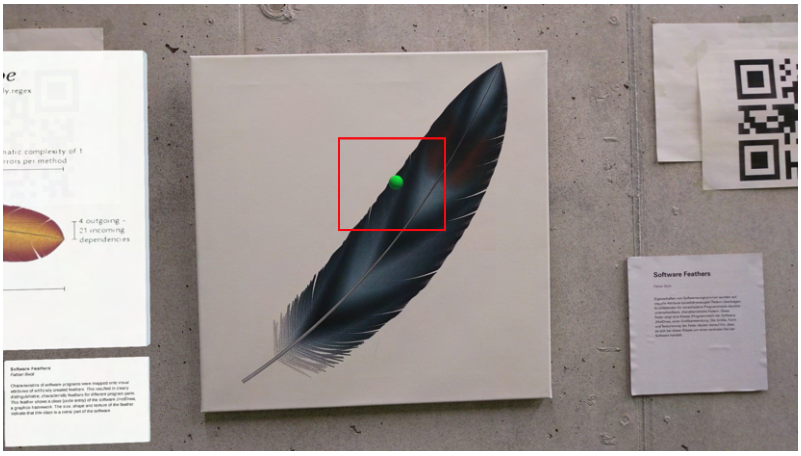


Figure 5.7: The green sphere in the image represents the gaze point, which is only visible in the video.

We chose ARETT for our pilot study because it provides a stable data rate for gaze recording and an easy-to-use interface. We did not use the control interface to start and stop the gaze recording, as the video recording should start at the same time. Therefore, we wrote a program that triggers the start of data and video recording with different keystrokes via a Bluetooth keyboard connected to the HoloLens. We utilized the interface to confirm the eye calibration and name the CSV file.
The integration of the WLT project stated in Section 5.1.3 caused one issue with the virtual scene implementation. The GazeProvider did not recognize the transformation of the coordinate space to the physical environment, hence the gaze position was not changed. The transformations are automatically detected and corrected when utilizing the functions accessible in MRTK. However, because ARETT accesses the gaze data via the native API, the DataProvider script in ARETT has to be modified by adding a transformation to the gaze data.

Because we had enabled spatial mapping in our pilot study, each gaze sample got a hit point. While the virtual objects that were hit could be identified by their names, the real objects that were seen could only be recognized as part of a mesh, necessitating manual annotation.

For the post-processing utilized in Chapter 6, we needed the following data from the recorded ARETT data:

- **EyeDataTimestamp**: contains the Unix timestamp of the gaze data in ms.

- **gazeHasValue**: is true if the eye was detected and a valid gaze data exists.
- **gazeOrigin_x, gazeOrigin_y, gazeOrigin_z**: contains the gaze origin's transformed position. Later in the gaze replay visualization, this will be used to display a ray starting from the gaze origin.
- **gazeDirection_x, gazeDirection_y, gazeDirection_z**: contains the gaze direction's transformed position. Later in the gaze replay visualization, this will be used to visualize a ray from the gaze origin to the gaze direction.
- **gazePoint_x, gazePoint_y, gazePoint_z**: contains the gaze point's transformed position in the global coordinate system.
- **gazePoint_target_name**: includes the name of the object that was hit. A virtual object is given the name generated by the AR scene. Because only the mesh of the spatial mapping is present for a real object, the name of the mesh is described.
- **GazePointWebcam_x, GazePointWebcam_y, GazePointWebcam_z**: contains the position of the projected gaze point on the virtual camera. The frame around the point (GazePointWebcam_x, GazePointWebcam_y) is extracted from the video in Chapter 6.2.

Because ARETT includes an R package for post-processing gaze data, we used it to extract fixations, as mentioned in Chapter 6.2. ARETT offers several approaches for identifying fixations. Fixation can be identified using a velocity threshold [Ols12], a dispersion threshold [LMGA20], or AOIs [KBM+]. In our case, the fixations were calculated by a velocity threshold. To do this, we had to first compute the velocity of each gaze point, which is also provided by an ARETT R package [KBM+], and then run the fixation algorithm.

The remaining steps for the visualization were performed using the calculated fixations, as detailed in Chapter 6.

6 Visualization Design

This chapter introduces our visualization tool. We go over the individual visualization components in depth and outline the interaction methods used. The final section discusses the steps involved in developing this tool.

6.1 Overview

Two visualization techniques are the main components of our visualization tool. These are the gaze replay visualization and the timeline visualization, which are both highlighted in Figure 6.1. They are linked together to visualize ARETT's spatio-temporal data. They are outfitted with a variety of UI components to display various information from the data. These are shown as toggle buttons on the right side of the GUI. There's also a scroll view with toggle buttons that provides a participant list.

Because data annotation is one of the primary functions of our visualization, there are two UI toolbars within the two visualization techniques to define AOIs and annotate data. An AOI legend with scroll view lists the defined AOIs. They are classified as virtual or real and assigned a color. Initially, 10 virtual AOIs were defined in the pilot study. ColorBrewer [Cyn10] is used to define a ColorMap for these. A random color is generated for the other AOIs, which will be identified later. The individual components of our visualization tool and their functions will be described in detail in the next section.

6.2 AOI Timeline Visualization

For our tool, we employ a timeline visualization to present frames of the fixated areas over time and produce annotations depending on the frames. The timeline visualization has time on the x-axis and participants on the y-axis. Fixations are obtained from each participants' gaze data and displayed in the timeline. A frame and a bar enclosing the frame represent a fixation (see Figure 6.2).

The fixated point's environment is fetched from the video to construct the frame. Since a fixation consists of several gaze data, the frame is created using the middle gaze point. If the number of gaze points is even, the rounded gaze index is applied.

The length and width of the frame represent the distance to the focused object. We normalize the distance by taking the longest distance of all participants. The further away the focused point is, the smaller the frame will be displayed. The largest frame has the size 70x70.

The frame is surrounded by a bar that indicates the start time and the duration of the fixation. The length of the bar specifies the duration of the fixation. The height corresponds to the size of the frame, which means that the distance of the focused point is displayed here.

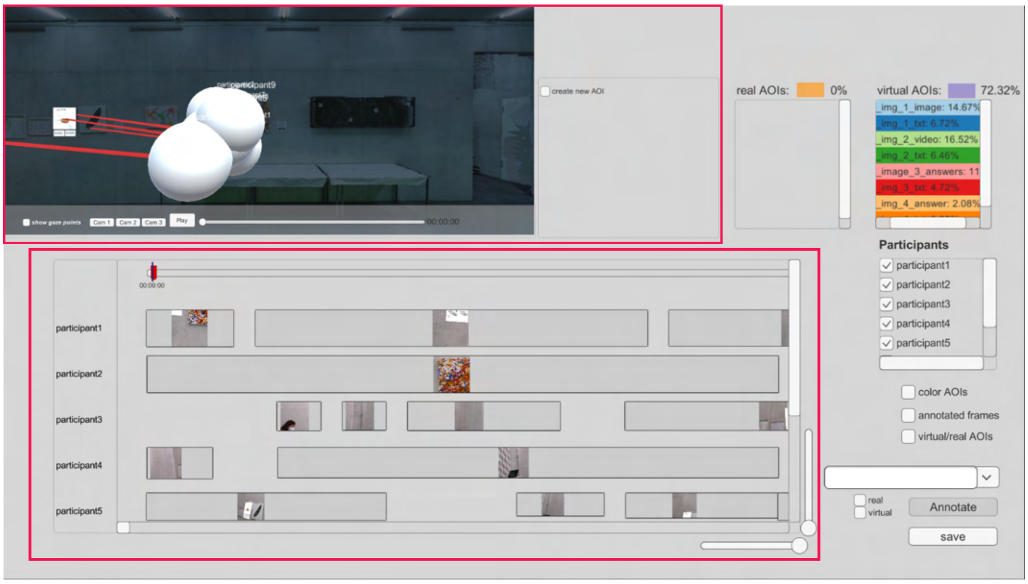


Figure 6.1: The GUI of the visualization tool is shown. The highlighted region at the top depicts the gaze replay with the AOI definition tool bar. The timeline visualization is shown in the highlighted area at the bottom. Several UI elements on the right side cause changes in the two visualizations.

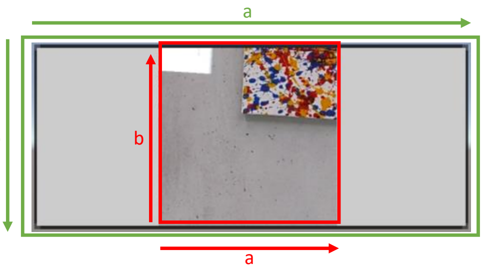


Figure 6.2: The red box surrounds the frame, and the green box the frame duration bar; combined, they represent a fixation. a. and b. (red) depend on the distance to the hit gaze point. The length of a. (green) represents the length of fixation, and b. (green) contains the same data as a. and b. (red).

The frames are positioned sequentially from left to right in the timeline. The positions of the frames for an individual participant vary in the x-direction, they are determined by their fixation time. The fixation bars' positions are determined by the start time of their fixation. If there is invalid data between fixations, there will be gaps between the frame bars.

Since we have only one global timeline, there is synchronization between participants. Figure 6.1 illustrates this, with each row representing one participant's fixation sequence. We observe that the fixations of individual participants occur at different time intervals.

6.3 Gaze Replay Visualization

We integrate a gaze replay into our visualization to provide spatial context to the 3D gaze data. For this, we reconstruct our AR scene by incorporating a photogrammetry mesh, created with RealityCapture, and a spatial mesh generated by the HoloLens during the study, into the visualization. Because the spatial mesh is not textured, we require the textured photogrammetry mesh.
As only the spatial mesh's alignment is identical to our AR scene, the photogrammetry mesh must be manually scaled, translated, and rotated to match. However, since our photogrammetry mesh lacks a mesh collider, we use the spatial mesh's collider.
In gaze replay, the recordings from the study are simulated by representing each participant as a sphere and visualizing their gaze behavior with a gaze ray. Here, the spheres have a floating label with the participant's name to distinguish them from each other. If we press the *Start* button, we can observe how the individual participants move and where they look. Thereby, we can use a timeline to jump to different times. The timeline of the gaze replay is linked to the timeline visualization. Thus, moving the slider in the gaze replay to that time also moves the slider in the timeline visualization.

The toolbar of the gaze replay also contains the toggle button *show gaze point*. When this is turned on, the participants' gaze points are mapped into the scene as small spheres. We can track the viewed regions by moving the slider slowly to define AOIs later at the regions with gaze points.

6.4 Interaction

For a better understanding and handling of the data, we have implemented interaction techniques in our visualization tool that fall into some of the 7 categories defined by Yi et al [YKSJ07]. These are presented below.

6.4.1 Consider different Views

In our visualization tool, different views can be examined to analyze the data and search for visible patterns in overviews. The following summarizes the interaction approaches employed, which fall under the Explore and Abstract/Elaborate categories.

Timeline Visualization It is not possible to display all of the frames because the timeline visualization creates hundreds of frames for each participant. Instead, the panning interaction technique can be used to explore the timeline. The user can explore the data by dragging the view or moving the scroll bar in the timeline visualization. The timeline can be used to identify which time span is currently being displayed. The timeline visualization also provides an overview of the data by zooming out. Zooming can be done in two directions: horizontally and vertically. They can be applied using the sliders that are located in the lower right corner. Vertical zooming is used if the user wants to consider all participants in the view. Horizontal zooming is used if the user wants all frames to appear in the view. Of course, this no longer allows for the identification of individual frames, but it does provide an overview of the annotated data when combined with the other methods outlined in Section 6.4.3 and

Section 6.4.4. Moreover, it facilitates finding patterns in the data.

It is possible to zoom in to see a smaller amount of data in an enlarged view. The user can zoom in and out using the scroll wheel in addition to the sliders.

Gaze Replay Since our visualization tool contains a 3D scene, it is necessary to use an Explore interaction technique. Therefore, three different views are available for the gaze replay. These can be selected using the *Camera1*, *Camera2*, and *Camera3* buttons on the toolbar. The default view is *Camera1*, which is implemented as a FreeLook camera where the user can freely move around the scene. Also in this view, when a frame is selected in the timeline visualization, it is pointed in the direction of the focused area to identify the selected frame faster in the gaze replay. The latter two cameras provide a static view of a subregion of the scene.

6.4.2 Identify and annotate AOIs

Interaction is essential for identifying and annotating AOIs. Following are two approaches for AOI annotation that can be performed utilizing our visualization tool. These methods integrate interaction techniques from the Select, Connect, Explore, and Encoding categories.

Timeline Visualization Individual frames in the timeline visualization can be selected and annotated here. When the frame is clicked, it is highlighted by being overlaid with a green layer. It's important to note that the frame bars aren't selectable and hence can't be highlighted. This is because, if the view is enabled for color encoding of AOIs, as indicated in Section 6.4.4, it would be confusing to choose a bar and highlight it in green while neighboring bars that already have an annotation are colored in a similar color (see Figure 6.3b). There can also be a need to modify an AOI of an already annotated frame. In this case the color-coding of the bar would be hidden. If many frames are expected to have the same AOI, they can be selected in the timeline. The timeline can be browsed by panning to find these frames.

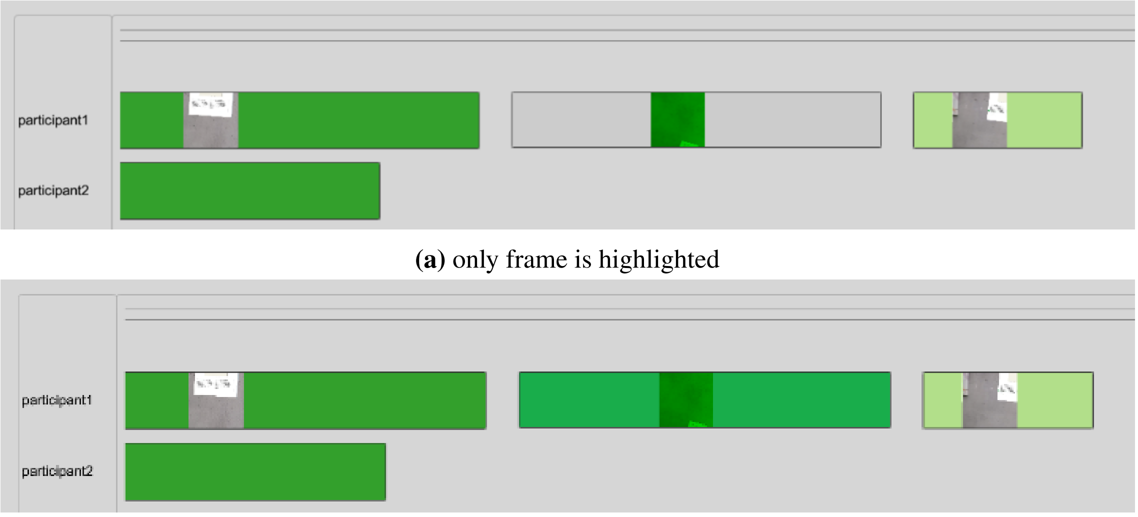
In order to provide global spatial context to the frames, an interaction technique that falls into the Connect category is needed here. Using the brushing and linking method, for each selected frame, the fixed region is highlighted in the gaze replay by mapping a green box to the fixed region. This allows the user to determine which region of the scene the frame belongs to. After selecting the frames, the selection box, which consists of an input box and a dropdown control, is used to define an AOI. If it is a completely new AOI, it is typed into the input box. If the AOI already exists, it can be selected from the dropdown control.

In addition, the toggle elements must be used to label whether the AOI is real or virtual. Multi labeling is also allowed for cases where it is not unique. Afterwards, the *Annotate* button is pressed to confirm the annotation.

When a new AOI is defined, it is added to the AOI legend and encoded with a randomly generated color.

Gaze Replay The brushing and linking interaction method can be used here by picking a region in the scene and then selecting all frames that have hit this region in the timeline visualization. They can be annotated there.

We must first enable the *create new annotation* checkbox in order to use this technique. This will bring up a new window with numerous alignment options for the AOI region. First, we name the AOI we want to construct. We can position a cube in the gaze replay once we select



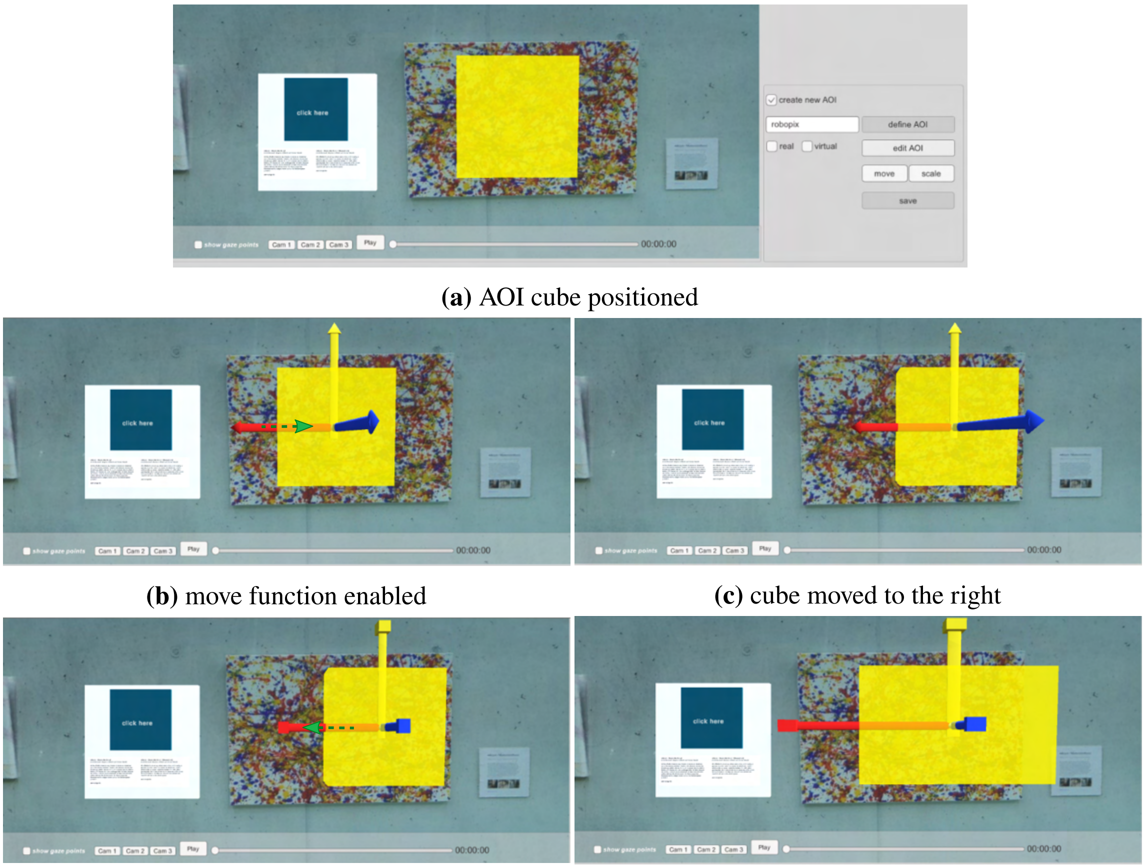
(a) only frame is highlighted

(b) frame and fixation duration bar is highlighted

Figure 6.3: Figure 6.3a shows how to select a frame. It would be confusing for the user if nearby frames with green-labeled AOI existed, as shown in Figure 6.3b.

define AOI. Here, an Explore interaction method is needed to access any region in the scene. Therefore, the *Camera1* view is turned on to be able to move freely in the scene.
In the beginning, the cube can only be positioned in the regions where the mouse has touched the mesh. However, we can change the cube after it has been placed in the gaze replay. When we select *Edit AOI*, we have the option to scale or move the cube. Alternatively, the cube can be deleted if, for example, the wrong AOI name was entered.
The user clicks on the corresponding button to activate one of the editing functions. After selecting the cube, three axes appear inside the cube. The axes are used to scale and move the cube. Depending on the desired direction of scaling or movement, the corresponding axis is used. To do this, click on the axis, hold it down and move it in the corresponding direction. When scaling, move the mouse along the axis towards the cube to make the cube smaller. Moving the mouse away from the cube increases the cube's size in the axis direction. The move function is the same. Depending on the axis chosen, moving the mouse towards the cube will move the cube to the right, down, or back (see Figure 6.4). When the AOI cube is in the desired position, the *Save* button is pushed, and all frames inside the designated region are selected in the timeline visualization, as the two visualizations are linked. There, the defined AOI must be selected from the drop down view and then the annotation for these frames must be typed using the *Annotate* button. If the defined AOI in the gaze replay does not already exist in the AOI legend, it will be added and a random color will be generated. Technically, this method can be used to annotate frames and define an unlimited number of AOIs. However, if there are too many AOIs, it will be difficult to compare them.

When the frames are annotated using the two methods described, the user can compare the individual AOIs and the virtual and real AOIs based on their ratio. This is displayed in the AOI legend and applies only to the selected participants. The calculation is performed depending on the fixation duration.



(a) AOI cube positioned

(b) move function enabled

(c) cube moved to the right

(d) scale function enabled

(e) cube scaled in x direction

Figure 6.4: Figure 6.4a shows how to define an AOI region. Once a cube is placed, it can be moved and scaled to take the position and size of the real AOI. In Figure 6.4b the red axis is moved in the direction of the green arrow, resulting in Figure 6.4c. The same applies to Figure 6.4d, where the red axis is used to scale the cube. Moving to the left makes the cube scale-out (see Figure 6.4e

6.4.3 Filter Data

An interaction technique from the filter category is available in our visualization tool. To filter participants from the study, we use a variant of dynamic query control [AWS92], e.g., to use only the data from participants that have a similar pattern concerning AOIs. Using the toggle buttons, the user can select the participants whose data should be examined from the participant list (see Figure 6.1). The visualization of the timeline and the gaze replay will then show only the data of the selected participants.

For the annotations, another filter technique is available. Using the *annotated frames* toggle button, all already annotated frames, including the fixation duration bars, are grayed out, leaving only the frames that need to be annotated highlighted, similar to [ST98]. At the same time, since the annotated frames are still visible, the context is not lost. The user gets an overview of the frames that need to be annotated by combining this method with the zooming function.

6.4.4 Comparison of AOIs

The color encoding we used in our visualization tool is included in the Encode category. After the frames have been labeled with an AOI, one of two color encoding methods can be enabled. The first method includes coloring all fixation duration bars with the color of the AOI specified in the AOI legend. A scarf plot is created as a result of this. The user gets an overview of the AOIs when combined with the zooming-out method. This overview shows the order of the AOIs and the duration of their fixation which can be examined in detail.

The color encoding of the grouped AOIs is considered in the second technique. The AOIs are categorized based on virtual and real AOIs, therefore the frames are only mapped to three colors (virtual, real, virtual&real). A scarf plot is created as a result of this. The user gets an overview of the virtual and real AOIs when combined with the zooming-out method. This overview shows the ratio of the two groups as well as the frequency of change between them.

6.5 Implementation

We used Unity 2020.3.24 for our visualization application, which is built on the Windows standalone platform with an x86-64 architecture. The project includes three scenes. Scene 1 extracts the data required for the visualization. Scene 2 shows the timeline visualization, while scene 3 shows the gaze replay. All scenes have access to the StreamingAssets folder, which contains the gaze and video data.

The first scene is loaded first to extract the fixations from the gaze data. The R library provided by ARETT can be used to calculate fixations based on the velocity threshold. To do this, a process is started via the System.Diagnostics.Process class that executes an R script.

The fixation data is used to extract frames from the videos recorded in the pilot study. Each fixation is made up of a set of gaze points. Because a frame is supposed to represent a fixation, only the gaze point in the middle of the list is used for the fixation. For each 3D gaze point, there is a projected 2D gaze point on the camera image, called gazePointWebcam. The timestamp and gazePointWebcam of the viewed gaze point are utilized to retrieve the frames. Using Unity's VideoPlayer class, the video goes to the timestamp for each fixation and extracts a 400x400 pixel frame around the gazePointWebcam.

After the data extraction is complete, the scene is unloaded, and Scenes 2 and 3 are launched sequentially to display the visualization.

In Scenes 2 and 3, all data in the StreamingAssets folder is dynamically loaded. If this folder contains the data of all study participants, the application will load all of them.

Scene 2 generates a timeline with frames from all participants, as well as the participant list and the AOI legend. Scene 3 depicts the spheres in gaze replay that represent the participants. The SceneDataManager component, which is included in both scenes, is used to connect the two scenes. This component has a MonoBehaviour script that only uses static variables to pass data between the two scenes. Thus, when the toggle buttons in the participant list are toggled on and off, the gaze replay is notified of the state change. When a participant is unselected, the generated GameObjects in both scenes are disabled but not destroyed.

Data persistence is provided by the visualization application. When a user annotates, creates AOI regions, or generates AOI labels, these are all saved in a .json file and loaded when the application is restarted. That is why there is a serializable script with data structures for the frames, AOI labels, and constructed AOI regions, so that this data may be saved in a .json file and read back.

7 Case Study

In order to evaluate the usability of our visualization technique for the annotation of real AOIs as well as the analysis of real and virtual AOIs, we collected data in a pilot study and analyzed it in our visualization. In the following, we first describe the pilot study. Then, we present the results of the case study.

7.1 Pilot Study

The pilot study was conducted with 10 people consisting of employees from the Visualization Institute of the University of Stuttgart (VISUS) and students. The basement of the VISUS building was used for the study. There are 6 images hanging in the corridor (length about 12 m, width about 6 m). These images were used in the study.

AR scene In the AR scene, there was a virtual board next to each image (see Figure 7.1). Their content varied, and some of them were interactive and dynamic, allowing different use cases to be considered in the analysis. Each board had a description of the related image. Table 7.1 lists the content of the boards and assigns them to stimulus types.

Study procedure The study procedure can be divided into two parts. The first half includes the preparations for the actual study. This includes the participant reading and signing the consent form. If the participant has no previous experience with the HoloLens, a brief introduction is given. After the participants put on the HoloLens and walked to the marked start position, they performed an eye calibration of the HoloLens.

The second part consists of the actual study. The AR application is launched on the HoloLens. The participants task is to solve 5 questions. To do this, they walked to the 5 images in turn and followed the instructions of the study conductor. For each image, the participants are asked to read the image description (see Figure A.6). Then they are asked to look at the rest of the virtual board and the picture next to it. A simple question is asked about the content being viewed. After the participants give a meaningful answer, they can move on to the next image. The individual tasks are numbered from 1 to 5. In Figure 7.2 the viewed images and virtual boards are shown. Their positions in the scene are displayed in Figure 7.1.
The individual tasks are set as follows:

Task 1: *"Next to the image is a virtual board. Below is the description of the image. Read it briefly. let me know when you are done. The labeled image on the board is another program part of the same software. Compare the two feathers. Name two differences based on the labeling in the virtual image.*"(see Figures 7.2a and 7.2b)

Task 2: *“Briefly read the description for the second image on the virtual board. Above the description is a short video, click on the video by pointing with your hand and making the click gesture. Is the painting in the video and the real painting the same?”* (see Figures 7.2c and 7.2d)

Task 3: *“Again, read the description briefly. You can see another input image on the virtual board. Select the appropriate hough transform for that image. Click on the answer. Click directly on the virtual image.“* (see Figures 7.2e and 7.2f)

Task 4: *"Here is it not necessary to read the description. Look at the image briefly and guess what the answer to the question on the virtual board is.“* (see Figures 7.2g and 7.2h)

Task 5: *"Read the description briefly. Which features of the face are easy to recognize which parts are more difficult to recognize?“* (see Figures 7.2i and 7.2j)

In Table 7.1 is a short overview of the virtual content of each task. The order in which the pictures were viewed varied. 5 participants went clockwise from task 1 to task 5, 5 participants went counterclockwise.
In total, the study took about 15 minutes.

Table 7.1: Overview of the virtual boards and their contents. The ID corresponds to the labeling in Figure 7.1

Task	description	stimuli type
1	labelled image, description text of image	static
2	video of how robopix image is created, description text of image	dynamic
3	question with selectable images as response, description text of image	active, dynamic
4	question with selectable boxes as response, description text of image	active, dynamic
5	picture, description text of image	static

7.2 Data Exploration

With the recorded gaze data from the pilot study, we used our visualization tool to annotate real AOIs and then investigated the distribution of fixations on the different AOIs.
Of the data recorded, one participant had the lowest percentage of valid data at 86.48%, and one participant had the highest percentage at 95.24%. We did not discard any participants due to invalid data. However, there was a problem in the postprocessing step with one participant where the fixation algorithm in RStudio threw an error. Since the reason was unknown, this participant was not included in the analysis.

7.2.1 Annotation technique

Our visualization tool provides multiple functions to support annotations. They have already been described in the Chapter 6. We used several strategies to make the annotation as efficient as possible. A total of 6252 frames were generated from our data, of which 4509 frames were already annotated, as virtual AOIs were automatically identified when the gaze hit on virtual objects. The initial state

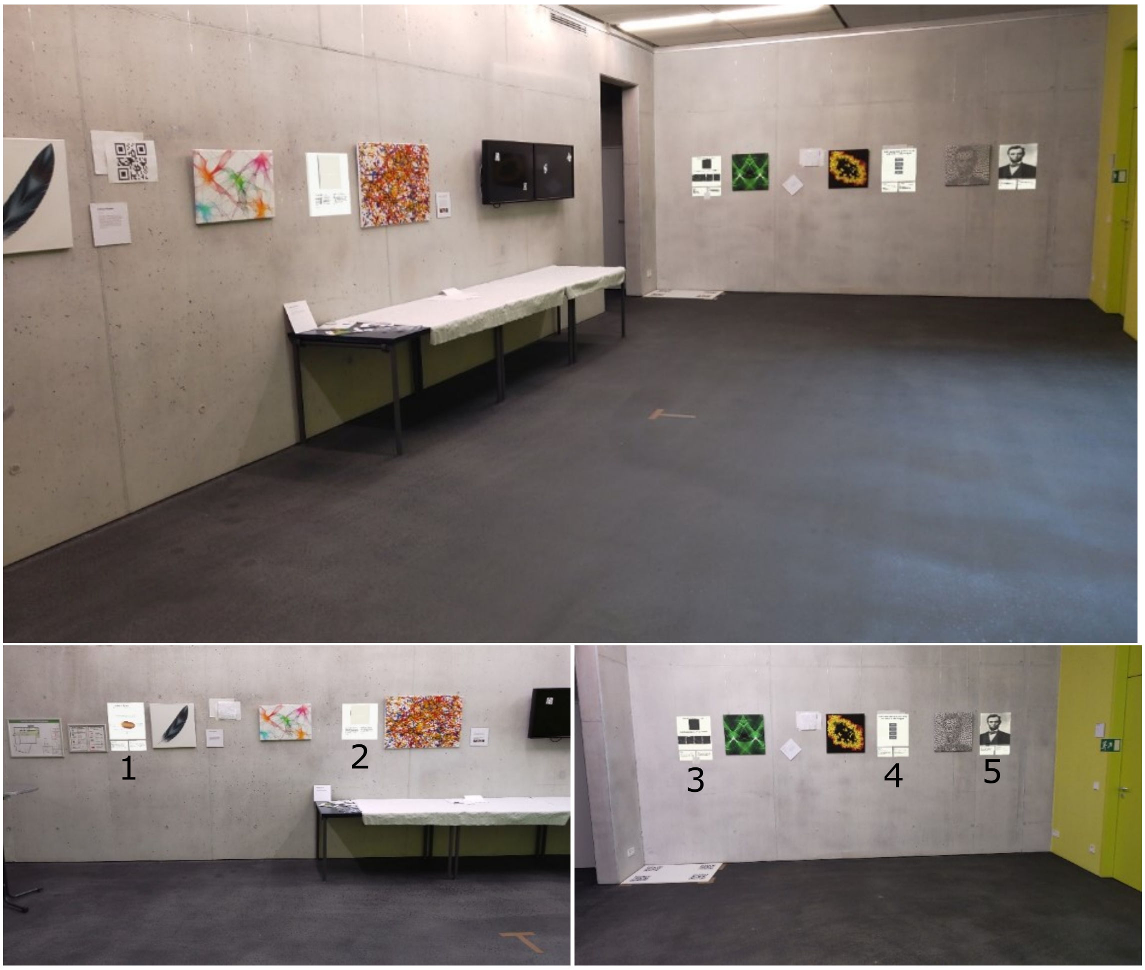
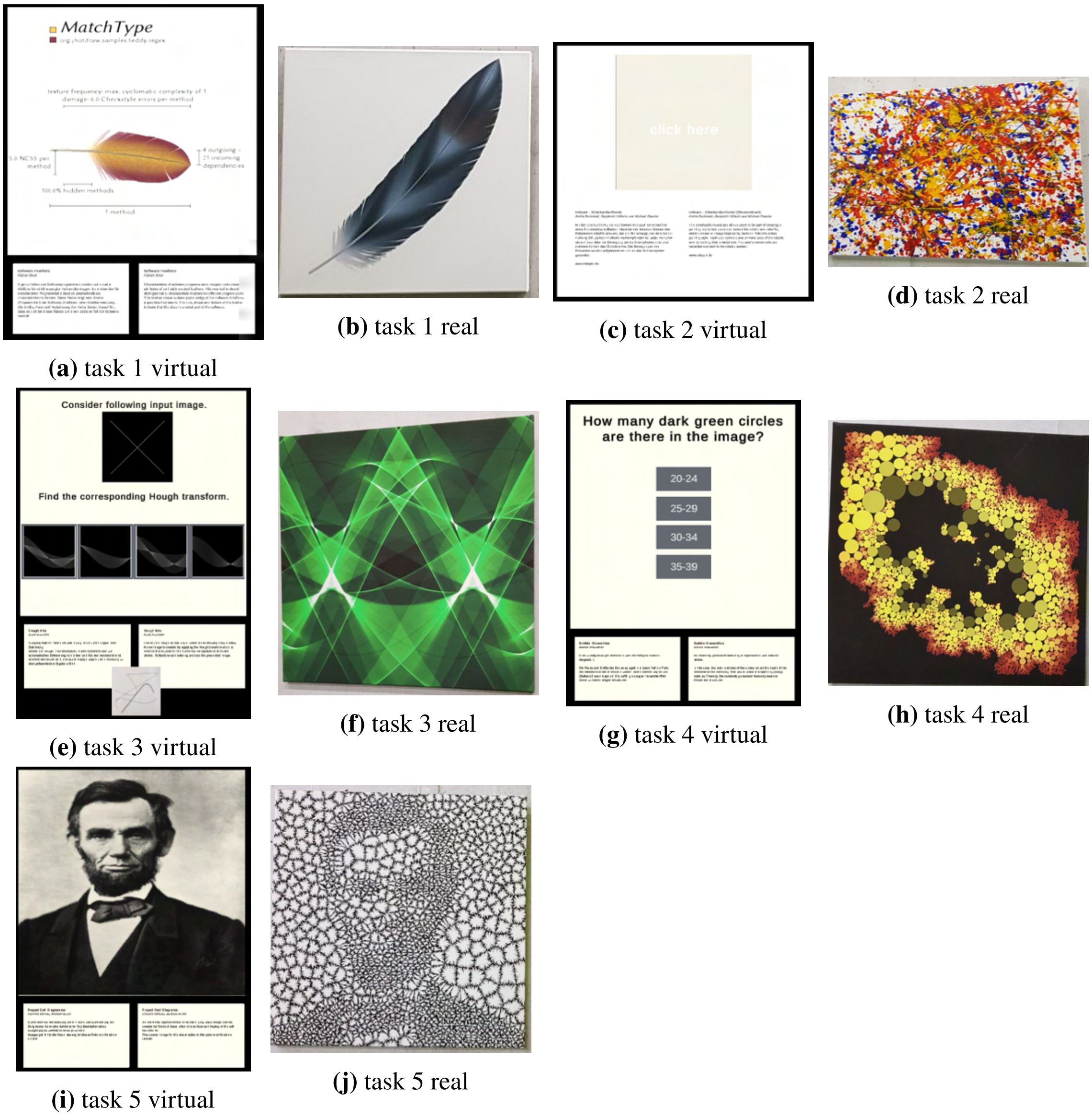


Figure 7.1: Different views of the AR scene taken in the basement. The bottom two images are labeled with numbers representing the positions of the tasks.

of our visualization with the collected data is shown in Figure 7.3. Knowing the AR scenario, we can first define the regions of real AOIs in the gaze replay. The number of frames found for the known 5 real AOIs (i.e., the real part of the tasks in Figure 7.2) varied:

- real_img_1: 293 frames,
- real_img_2: 282 frames,
- real_img_3: 194 frames,
- real_img_4: 233 frames,
- real_img_5: 376 frames.

In total, 1378 frames were annotated using this method.

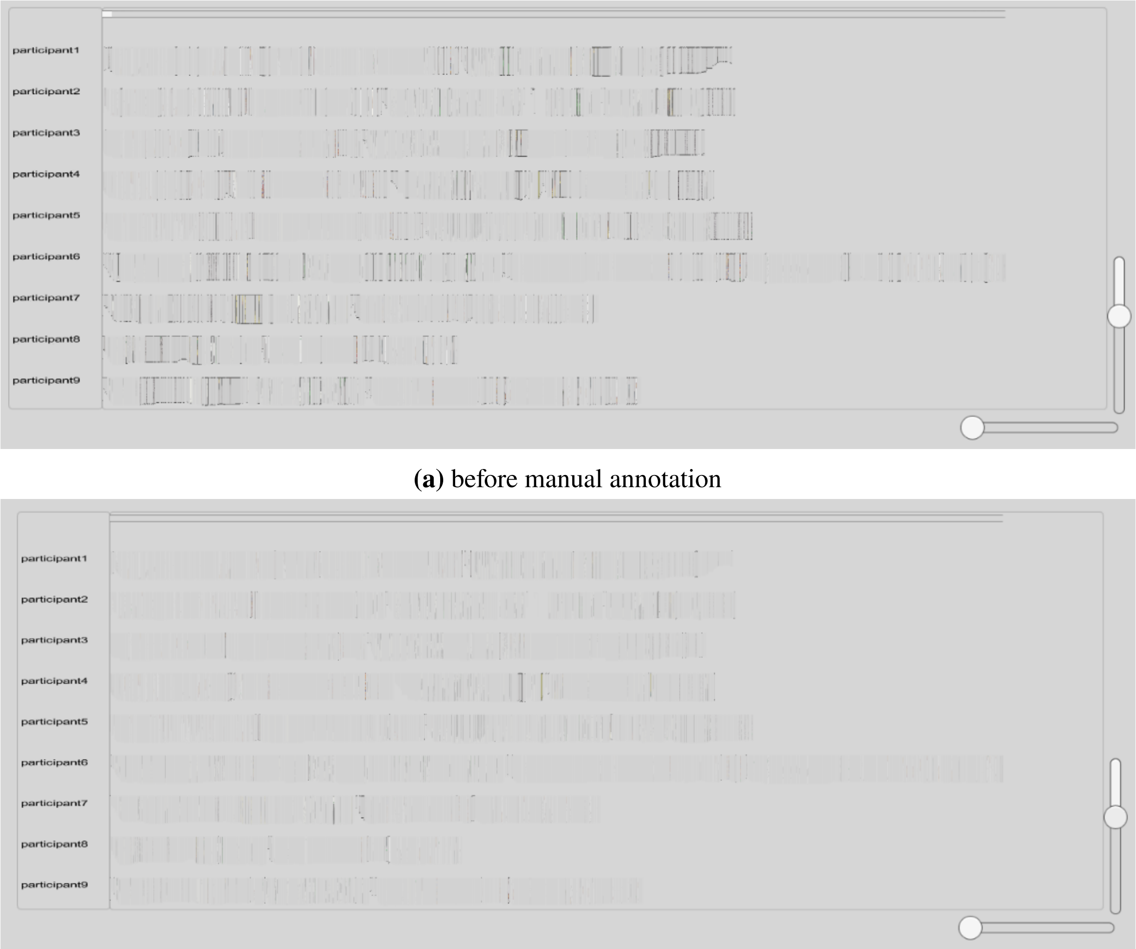

MatchType
click here
Consider following input image.
Find the corresponding Hough transform.
How many dark green circles are there in the image?
20-24
25-29
30-34
35-39

(a) task 1 virtual **(b)** task 1 real **(c)** task 2 virtual **(d)** task 2 real **(e)** task 3 virtual **(f)** task 3 real **(g)** task 4 virtual **(h)** task 4 real **(i)** task 5 virtual **(j)** task 5 real

Figure 7.2: The real images and virtual boards viewed in each task are shown.

We noticed that several of the virtual AOIs were not identified during annotation. This can happen if the vertical alignment of the HoloLens mesh against the walls is not smooth. We identified the regions of these frames in the gaze replay and re-defined AOIs in that area. After using the gaze replay to define real and virtual AOIs, 82.53% of the frames were marked as virtual AOIs and 15.49% as real AOIs. In total, 98.02% of the frames were annotated.

For annotating the remaining frames, we used different combinations of the provided features in our visualization tool. First, we grayed out annotated frames to identify frames that were not yet assigned to an AOI. Figure 7.3 shows the timeline visualization with the feature applied. In this visualization, the frames that had not yet been labeled were easy to detect. Since the timeline was very long, manual searching and labeling was tedious.



(a) before manual annotation

(b) after manual annotation

Figure 7.3: A timeline visualization with greyed out feature is shown. The timeline visualization in Figure 7.3a contains only the data from the pilot study before applying the annotation step. Figure 7.3b displays the data after applying the annotation step.

Another helpful method was to color the already annotated frames to find the remaining unlabeled frames. If a frame was located between two frames with the same AOI, the context was easier to identify. From this, it could be assumed that the frame in between also had the same AOI.

It was difficult to match some frames with the gaze replay. There were frames where the area showed did not match the frame displayed. These were marked as unknown. We can see in gaze replay some gaze points hanging in the air. The reason for this was that when the participants turned to face the conductor, their gaze fell on the conductor's mesh.

In several iterations, the timeline visualization was zoomed out, searched for the remaining unmarked frames and zoomed in again.
It took about 1h 20min to annotate all of the frames found. 82.69% of the frames were defined as virtual AOI, 16.07% as real AOI, the rest were not defined because these fixations were between tasks.

7.2.2 Analysis of annotated data

For the analysis, a comparison was made between virtual and real AOIs, between individual AOIs, or between groups of AOIs belonging to the same task.

The formation of two groups can be noticed regarding the viewing order of the AOIs. Group 1 started looking at task 1 and walked clockwise along the frames until task 5. Group 2 looked at the frames counter-clockwise. (task 5 to task 1). Figure 7.4a shows that participants 1 to 5 belong to group 1. Participants 6 to 9 form the second group.

This behavior could be seen more clearly in gaze replay by looking at the start and end positions of each participant. The ratios of virtual and real AOIs for the different groups were as follows:

- Group 1: 14.02% real AOI, 76.33% virtual AOI
- Group 2: 18.84% real AOI , 80.02% virtual AOI

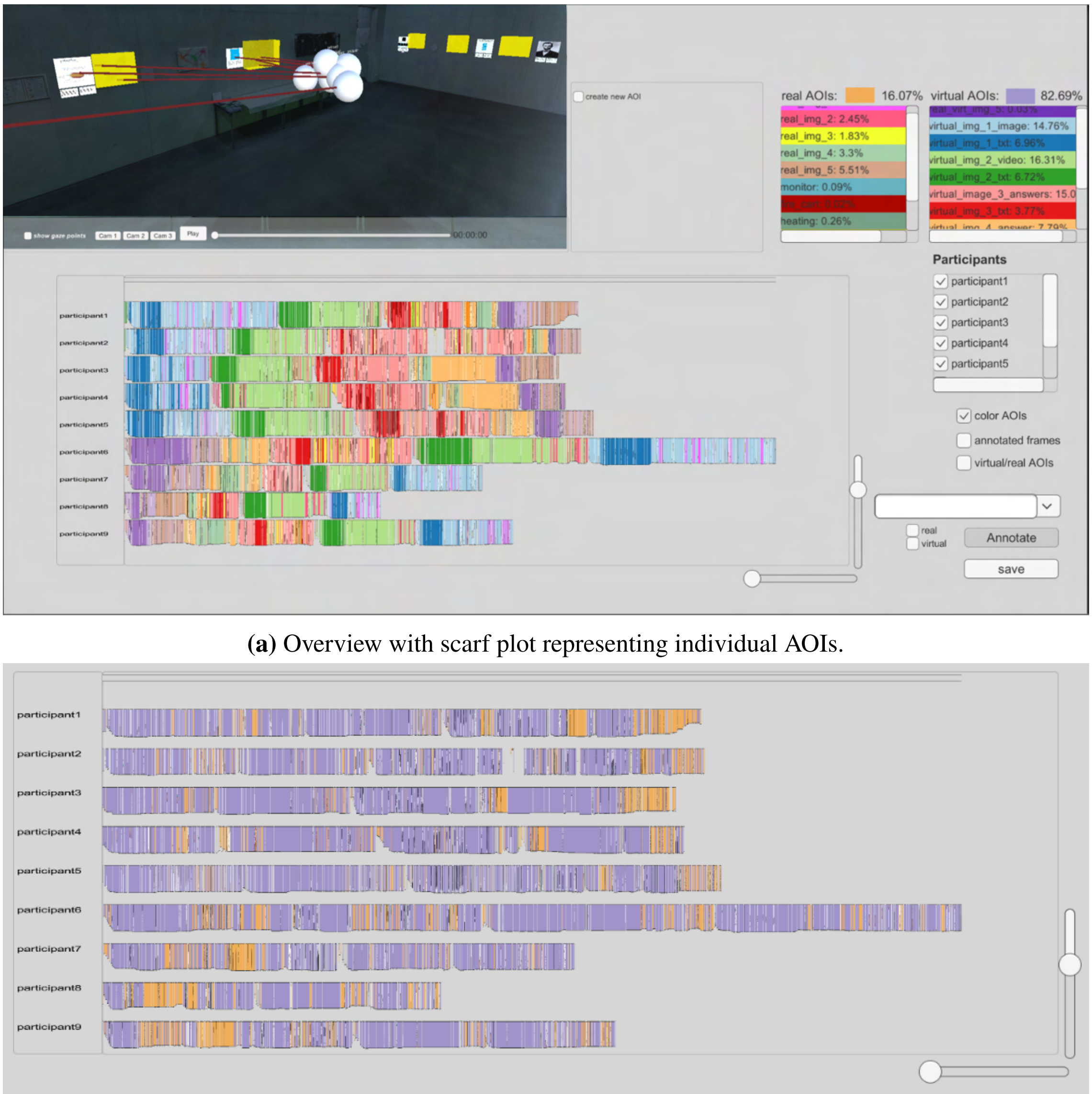
In the overview in Figure 7.4a, we noticed that participants 7, 8, and 9 had more real AOIs (22.0% real AOIs, 76.33% virtual AOIs) compared to the rest. This might be because participant 8 spent more time looking at real_img_5 (image next to task 5), and participant 9 and participant 7 focused longer on real_img_4 (image next to task 4). This can be observed in the two scarf plots (see Figure 7.5).

In the overview in Figure 7.4a, we noticed that participants 7, 8, and 9 had more real AOIs (22.0% real AOIs, 76.33% virtual AOIs) compared to the rest. This might be because participant 8 spent more time looking at real_img_5 (image next to task 5), and participant 9 and participant 7 focused longer on real_img_4 (image next to task 4). This can be observed in the two scarf plots (see Figure 7.5).

We can see that participant 6 had the longest session compared to the other participants (about 10min32sec). The viewer's attention was focused on the virtual img_5_txt, virtual img_2_txt, virtual img_1_txt, and virtual img_1_image AOIs, as seen in the scarf plot. This shows that the reading time for this group was longer. We observed that participant 6 had the longest session compared to the other participants (about 10min32sec). The viewer's attention was focused on the virtual AOIs ïmg_5_txt", "virtual img_2_txt", "virtual img_1_txt", and "virtual img_1_image", as shown in the scarf plot. This indicates that the reading time took longer for this participant.

The study period for participant 8 was shortest, taking only 4min10sec. One possible reason could be that this viewer was able to complete interactive tasks such as selecting responses on virtual boards quickly. For this, we looked at the AOIs representing task 4, task 3, and task 2 (see Table 7.1). The remaining tasks (task 4, task 5) were also completed quickly. In contrast, we saw in the scarf plot that participants 3 and 4 took longer to complete the tasks with interaction (task 3, task 4).

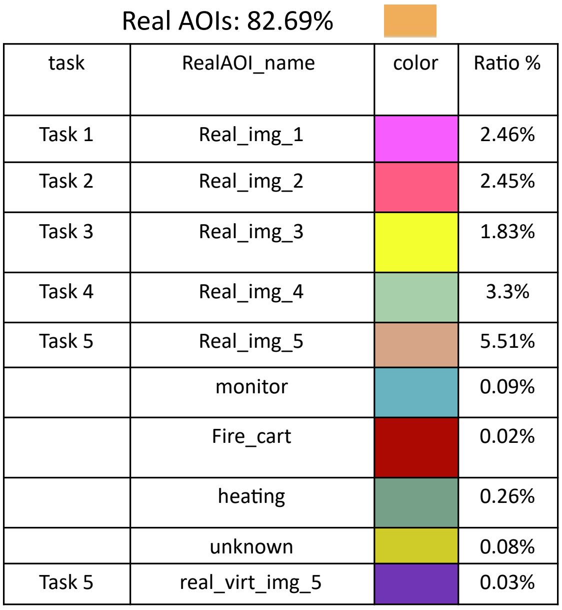
Comparing the individual AOIs, we found that the virtual_img_1_image and the virtual_img_2_video received the longest attention from all viewers. In the pilot study, we asked participants in task 1 first to read the text and then to view the virtual and real images. We asked for a simple comparison between the two images (the images were from the software Feathers [Bec14]). The participants found the comparison challenging, as seen by the scarf plot. Virtual_img_2_video belongs to task 2. Here, participants had to watch a short video. Since all participants finished watching the video, this could be seen by the length of this area in the scarf plot.



(a) Overview with scarf plot representing individual AOIs.

(b) Scarf plot represents the virtual and real AOIs.

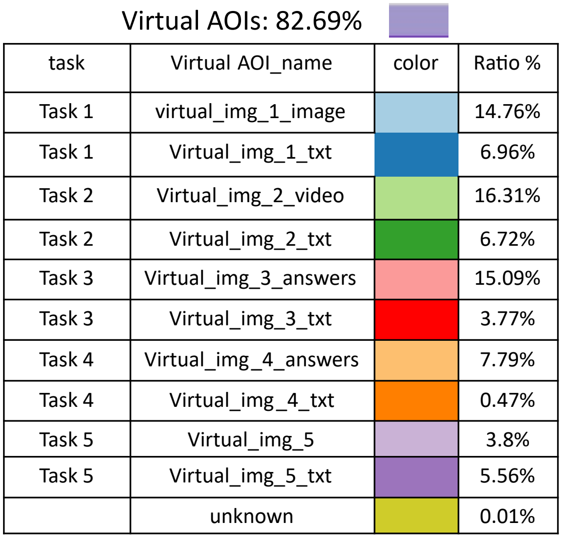
Figure 7.4: Figure 7.4a depicts our visualization after completing the annotation phase. The defined AOI areas are represented by the yellow boxes in the gaze replay. Individual AOIs are colored differently in the timeline visualization. Figure 7.4b shows the timeline visualization considering virtual and real AOIs.



Real AOIs: 82.69% 🟧

task	RealAOI_name	color	Ratio %
Task 1	Real_img_1	🟪	2.46%
Task 2	Real_img_2	🟥	2.45%
Task 3	Real_img_3	🟨	1.83%
Task 4	Real_img_4	🟩	3.3%
Task 5	Real_img_5	🟫	5.51%
	monitor	🟦	0.09%
	Fire_cart	🟥	0.02%
	heating	🟩	0.26%
	unknown	🟨	0.08%
Task 5	real_virt_img_5	🟪	0.03%

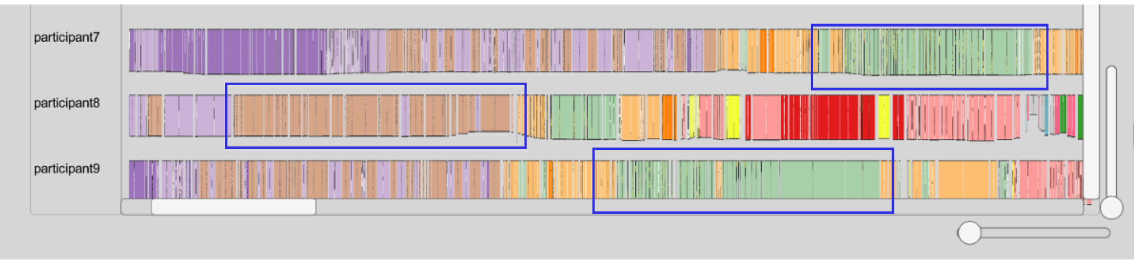
(a) legend of the labeled real AOIs.



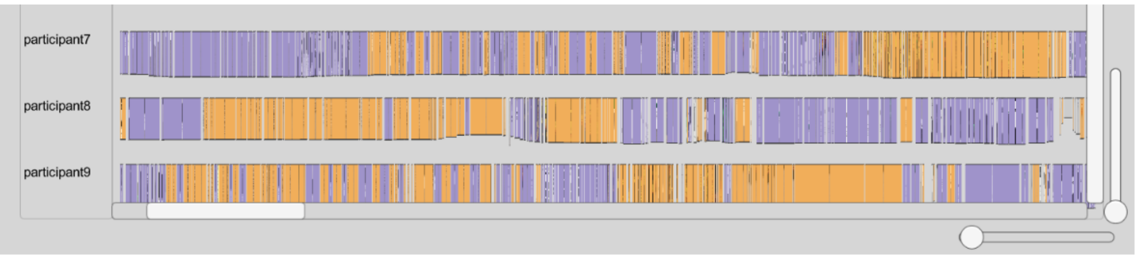
Virtual AOIs: 82.69% 🟪

task	Virtual AOI_name	color	Ratio %
Task 1	virtual_img_1_image	🟦	14.76%
Task 1	Virtual_img_1_txt	🟦	6.96%
Task 2	Virtual_img_2_video	🟩	16.31%
Task 2	Virtual_img_2_txt	🟩	6.72%
Task 3	Virtual_img_3_answers	🟥	15.09%
Task 3	Virtual_img_3_txt	🟥	3.77%
Task 4	Virtual_img_4_answers	🟧	7.79%
Task 4	Virtual_img_4_txt	🟧	0.47%
Task 5	Virtual_img_5	🟪	3.8%
Task 5	Virtual_img_5_txt	🟪	5.56%
	unknown	🟨	0.01%

(b) legend of the labeled virtual AOIs.



(c) Scarf plot represents individual AOIs.



(d) scarf plot represents real and virtual AOIs.

Figure 7.5: Scarf plots of the participants 7,8 and 9 with two different views. Coloring of the scarf plots can be taken from the legend in Figure 7.5b. The highlighted areas show that participant 8 spent more time looking at real_img_5. Participant 9 and 7 spent more time looking at real_img_4.

If we consider only real and virtual AOIs, we see that among the real AOIs, real_img_5 has received the attention for the longest time. This belongs to task 5, where the task was to read the text and then compare the real image consisting of a Voronoi tessellation of the Abraham Lincoln portrait, with the virtual image containing the original portrait photo. It should be considered which areas of the portrait were particularly well represented by the Voronoi tessellation. In the scarf plot, we could see how participant 1 walked into distance from the two images to compare them. For participants 3 and 5, we could also observe a slight curve in the plot, indicating similar behavior.

The curves appeared in different regions of the scarf plot. At the beginning of the study, we could see a growing curve. This was caused by participants starting in the middle of the scene and then moving toward the first image (task 1 and task 5, respectively). As the two elements are further apart between task 2 and task 3, the frames go from small to large again (see Figure 7.6).

With our visualization tool, we could examine the gaze behavior of multiple viewers in more detail and compare them with each other. For further analysis, we would benefit from the support of descriptive statistics.

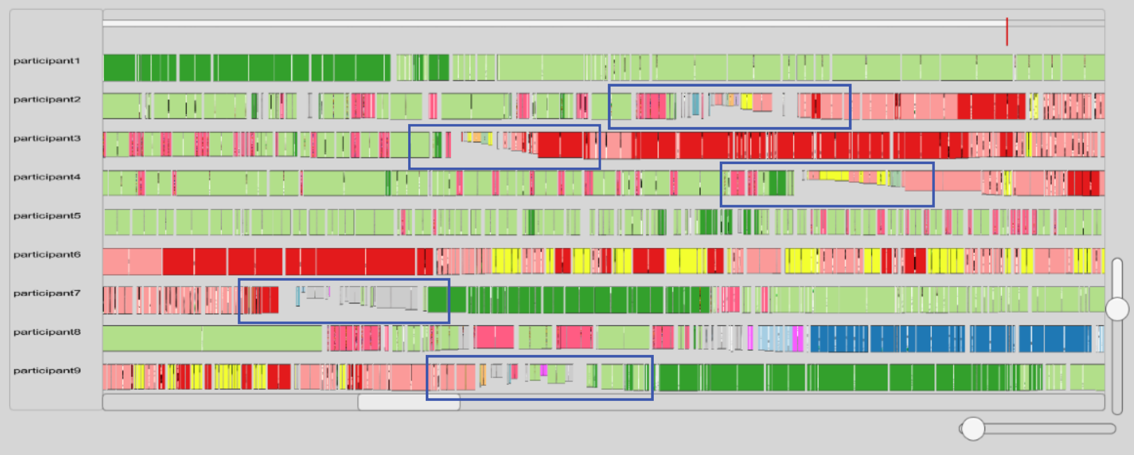


Figure 7.6: The scarf plot shows the transition between task 2 and task 3 where the participant has a longer path compared to the other tasks. As a resuk, the frames become smaller and a wave shape is created which can be seen in the highlighted area.

7.2.3 Limitations

The following summarizes the limitations of our visualization approach that we found in the case study:

- since the annotation of dynamic AOIs is not supported, frames with wrong spatial context are created in the gaze replay.
- if AOIs are defined that have been viewed only for a very short time, then they cannot be detected in the scarf plot because small frames are not visible due to horizontal zooming.

- During annotation, we noticed that some of the frames in the gaze replay were detected in a wrong region, even if the frame obviously depicted a real AOI. One possible reason for this is that the synchronization between the video recording and the gaze data recording was not perfect. Further investigation of the program is necessary to find out the cause.
- When there are too many AOIs defined, the scarf plot becomes distorted, and no meaningful comparisons can be made. A better comparison would be achieved by restricting the number of AOIs to ten or less.

8 Conclusion and Outlook

8.1 Conclusion

In this thesis, we created a visualization tool for analyzing gaze behavior in AR scenarios. We used the HoloLens 2 with an integrated eye tracker as hardware because it comes with Spatial Awareness. This enables the mapping of gaze data into a 3D world without the use of extra sensors.

We created a visualization tool for analyzing eye-tracking data using AOIs that combines two visualization techniques to produce a spatio-temporal context. The first visualization technique gives the temporal context. For this, we employed a timeline visualization that considers the participants' fixation data. Fixations are represented by frames that show the viewed gaze point, the distance between the gaze point and the viewer, and the duration of the fixation. These frames can then be labeled with AOIs.
A gaze replay visualization is used in the second visualization technique to convey spatial context. The behavior of the participants was simulated there using the spatial mesh offered by the HoloLens.

The two visualization techniques are linked so that information about both contexts is presented. They work together to identify virtual and real AOIs in order to do an AOI-based analysis later on. Because virtual AOIs are identified automatically, our approach focuses on identifying real AOIs. Therefore, both visualization approaches include a method for identifying and annotating the AOIs, which they achieve through their connection to each other.

For the analysis of AOIs, we created a scarfplot visualization to show the ratio of each defined AOI as well as the grouped AOIs by virtual and real AOIs.

We collected data in a pilot study and used it in the case study to build our visualization tool. With the implemented methods, we could annotate AOIs that we already know about. Our visualization did not detect dynamic AOIs. We were able to examine the considered order of each AOI from each participant and detect patterns while examining the annotated data. The AOIs could also be compared by real and virtual groups.

8.2 Outlook

For further analysis, additional features can be added to our visualization tool in future work:

- The defined AOI regions in the gaze replay can be colored according to their AOI as defined in the legend to explore them further. Because it has not yet been fully implemented in our visualization, we consider this method to be future work.

- Viewing and analyzing self-selected time intervals, for example, to evaluate only one specific task in the pilot study in detail.
- Implementation of additional filtering techniques, such as selecting the AOIs to be considered from the legend.
- Nested grouping of AOIs based on reading, image viewing, interaction, and other criteria, e.g. to compare reading behavior between real and virtual AOIs.
- Annotation of dynamic AOIs using an additional sensor to scan the environment and identify moving objects during the study. These will be featured in the gaze replay later on.
- By combining the visualization with a Space-Time Cube visualization, individual participants' walking behaviors can be taken into account.
- Combining the visualization with a heat map to examine the distribution of gaze points within AOIs in order to explain why an AOI was viewed for longer.

Incorporating these concepts into our visualization tool, as well as additional methods that consider the AR aspect, can be further explored in future work.

Bibliography

[AWS92] C. Ahlberg, C. Williamson, B. Shneiderman. “Dynamic queries for information exploration: An implementation and evaluation”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1992, pp. 619–626 (cit. on p. 40).

[Azu97] R. T. Azuma. “A survey of augmented reality”. In: *Presence: teleoperators & virtual environments* 6.4 (1997), pp. 355–385 (cit. on p. 13).

[Bec14] F. Beck. “Software Feathers figurative visualization of software metrics”. In: *2014 International Conference on Information Visualization Theory and Applications (IVAPP)*. 2014, pp. 5–16 (cit. on p. 48).

[BKR+16] T. Blascheck, K. Kurzhals, M. Raschke, S. Strohmaier, D. Weiskopf, T. Ertl. “AOI hierarchies for visual exploration of fixation sequences”. In: *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. 2016, pp. 111–118 (cit. on p. 15).

[BKR+17] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, T. Ertl. “Visualization of Eye Tracking Data: A Taxonomy and Survey: Visualization of Eye Tracking Data”. In: *Computer Graphics Forum* 36 (Feb. 2017). DOI: 10.1111/cgf.13079 (cit. on pp. 11, 13–15, 19).

[BSYB20] H. Bai, P. Sasikumar, J. Yang, M. Billinghurst. “A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing”. In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–13 (cit. on p. 18).

[Bur16] M. Burch. “A Task-Based View on the Visual Analysis of Eye Tracking Data”. In: July 2016. ISBN: 978-3-319-47023-8. DOI: 10.1007/978-3-319-47024-5_1 (cit. on p. 11).

[Car99] M. Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (cit. on p. 22).

[CN00] E. C. Crowe, N. H. Narayanan. “Comparing interfaces based on what users watch and do”. In: *Proceedings of the 2000 symposium on Eye tracking research & applications*. 2000, pp. 29–36 (cit. on p. 14).

[Cyn10] A. B. Cynthia. *ColorBrewer*. 2010. URL: www.ColorBrewer.org (cit. on p. 35).

[DLS+14] B. De Smet, L. Lempereur, Z. Sharafi, Y.-G. Guéhéneuc, G. Antoniol, N. Habra. “Taupe: Visualizing and Analyzing Eye-Tracking Data”. In: *Sci. Comput. Program.* 79 (Jan. 2014), pp. 260–278. ISSN: 0167-6423. DOI: 10.1016/j.scico.2012.01.004. URL: https://doi.org/10.1016/j.scico.2012.01.004 (cit. on p. 14).

[GH10] J. H. Goldberg, J. I. Helfman. “Comparing information graphics: a critical look at eye tracking”. In: *Proceedings of the 3rd BELIV’10 Workshop: BEyond time and errors: novel evaLuation methods for Information Visualization*. 2010, pp. 71–78 (cit. on p. 15).

[GK99] J. H. Goldberg, X. P. Kotval. “Computer interface evaluation using eye movements: methods and constructs”. In: *International journal of industrial ergonomics* 24.6 (1999), pp. 631–645 (cit. on p. 15).

[HNA+11a] K. Holmqvist, M. NYSTROM¨, R. ANDERSSON, R. DEWHURST, H. JARODZKA, J. VAN DE WEIJER. *Eye Tracking : A Comprehensive Guide to Methods and Measures*. English. United Kingdom: Oxford University Press, 2011. ISBN: 9780199697083 (cit. on pp. 11, 13).

[HNA+11b] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, J. Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. OUP Oxford, 2011 (cit. on p. 14).

[Hol01] J. Holsanova. *Picture viewing and picture description: Two windows on the mind.* Vol. 83. Lund University, 2001 (cit. on p. 14).

[HTA+18] K. Hagihara, K. Taniguchi, I. Abibouraguimane, Y. Itoh, K. Higuchi, J. Otsuka, M. Sugimoto, Y. Sato. “Object-wise 3D Gaze mapping in physical workspace”. In: *Proceedings of the 9th Augmented Human International Conference*. 2018, pp. 1–5 (cit. on p. 18).

[HWH19] H. Ham, J. Wesley, H. Hendra. “Computer Vision Based 3D Reconstruction : A Review”. In: *International Journal of Electrical and Computer Engineering (IJECE)* 9 (Aug. 2019), p. 2394. DOI: 10.11591/ijece.v9i4.pp2394-2402 (cit. on p. 15).

[KBM+] S. Kapp, M. Barz, S. Mukhametov, D. Sonntag, J. Kuhn. *ARETT-R-Package*. URL: https://github.com/AR-Eye-Tracking-Toolkit/ARETT-R-Package (cit. on p. 33).

[KBM+21] S. Kapp, M. Barz, S. Mukhametov, D. Sonntag, J. Kuhn. “ARETT: Augmented Reality Eye Tracking Toolkit for Head Mounted Displays”. In: *Sensors* 21.6 (Mar. 2021), p. 2234. ISSN: 1424-8220. DOI: 10.3390/s21062234. URL: http://dx.doi.org/10.3390/s21062234 (cit. on pp. 29, 31, 32).

[KBPW15] K. Kurzhals, M. Burch, T. Pfeiffer, D. Weiskopf. “Eye Tracking in Computer-Based Visualization”. In: *Computing in Science & Engineering* 17.5 (2015), pp. 64–71. DOI: 10.1109/MCSE.2015.93 (cit. on p. 11).

[KHH+15a] K. Kurzhals, M. Hlawatsch, F. Heimerl, M. Burch, T. Ertl, D. Weiskopf. “Gaze Stripes: Image-Based Visualization of Eye Tracking Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 22 (Nov. 2015), pp. 1–1. DOI: 10.1109/TVCG.2015.2468091 (cit. on pp. 11, 19).

[KHH+15b] K. Kurzhals, M. Hlawatsch, F. Heimerl, M. Burch, T. Ertl, D. Weiskopf. “Gaze Stripes: Image-Based Visualization of Eye Tracking Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 22 (Nov. 2015), pp. 1–1. DOI: 10.1109/TVCG.2015.2468091 (cit. on p. 19).

[KHW14] K. Kurzhals, F. Heimerl, D. Weiskopf. “ISeeCube: Visual analysis of gaze data for video”. In: Mar. 2014, pp. 43–50. DOI: 10.1145/2578153.2578158 (cit. on pp. 11, 14, 19).

[KPM+20] M. Kraus, T. Pollok, M. Miller, T. Kilian, T. Moritz, D. Schweitzer, J. Beyerer, D. Keim, C. Qu, W. Jentner. "Toward mass video data analysis: Interactive and immersive 4D scene reconstruction". In: *Sensors* 20.18 (2020), p. 5426 (cit. on p. 18).

[KSK13] I. Krejtz, A. Szarkowska, K. Krejtz. "The effects of shot changes on eye movements in subtitling". In: (2013) (cit. on p. 15).

[KW13] K. Kurzhals, D. Weiskopf. "Space-Time Visual Analytics of Eye-Tracking Data for Dynamic Stimuli". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2129–2138. DOI: 10.1109/TVCG.2013.194 (cit. on p. 19).

[LMGA20] J. Llanes-Jurado, J. Marín-Morales, J. Guixeres, M. Alcañiz. "Development and Calibration of an Eye-Tracking Fixation Identification Algorithm for Immersive Virtual Reality". In: *Sensors* 20.17 (2020). ISSN: 1424-8220. DOI: 10.3390/s20174956. URL: https://www.mdpi.com/1424-8220/20/17/4956 (cit. on p. 33).

[Mica] Microsoft. *Coordinate systems*. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/design/coordinate-systems#mixed-reality-experience-scales (cit. on p. 27).

[Micb] Microsoft. *Eye tracking on HoloLens 2*. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking (cit. on p. 30).

[Micc] Microsoft. *Framework and runtime*. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/architecture/framework-and-runtime (cit. on p. 27).

[Micd] Microsoft. *MixedReality-WorldLockingTools-Samples*. URL: https://github.com/microsoft/MixedReality-WorldLockingTools-Samples/tree/master/Advanced/QRSpacePins (cit. on p. 29).

[Mice] Microsoft. *Profiles*. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/profiles/profiles?view=mrtkunity-2022-05 (cit. on p. 27).

[Micf] Microsoft. *The basic idea*. URL: https://docs.microsoft.com/en-us/mixed-reality/world-locking-tools/documentation/concepts/basicconcepts (cit. on p. 29).

[Micg] Microsoft. *World locking and spatial anchors in Unity*. URL: https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/spatial-anchors-in-unity?tabs=wlt (cit. on p. 28).

[Mich] Microsoft. *World Locking Tools concepts*. URL: https://docs.microsoft.com/en-us/mixed-reality/world-locking-tools/documentation/concepts (cit. on p. 28).

[ML18] Z. Ma, S. Liu. "A review of 3D reconstruction techniques in civil engineering and their applications". In: *Advanced Engineering Informatics* 37 (2018), pp. 163–174. ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2018.05.005. URL: https://www.sciencedirect.com/science/article/pii/S1474034617304275 (cit. on p. 15).

[NK19] B. Nicoll, B. Keogh. "The unity game engine and the circuits of cultural software". In: *The Unity game engine and the circuits of cultural software*. Springer, 2019, pp. 1–21 (cit. on pp. 26, 27).

[OKYB21] S. Oishi, K. Koide, M. Yokozuka, A. Banno. “4D Attention: Comprehensive Framework for Spatio-Temporal Gaze Mapping”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7240–7247 (cit. on p. 18).

[Ols12] A. Olsen. “The Tobii IVT Fixation Filter Algorithm description”. In: 2012 (cit. on p. 33).

[ope] openmvs[AT]googlegroups.com. *OpenMVS: open Multi-View Stereo reconstruction library*. URL: https://github.com/cdcseacave/openMVS (cit. on p. 18).

[PB06] A. Poole, L. Ball. “Eye tracking in human-computer interaction and usability research: Current status and future prospects”. In: Jan. 2006, pp. 211–219 (cit. on pp. 13, 14).

[PDP+16] J. Pieszala, G. Diaz, J. Pelz, J. Speir, R. Bailey. “3D gaze point localization and visualization using LiDAR-based 3D reconstructions”. In: Mar. 2016, pp. 201–204. DOI: 10.1145/2857491.2857545 (cit. on p. 17).

[PKP10] D. Pontillo, T. Kinsman, J. Pelz. “SemantiCode: using content similarity and database-driven matching to code wearable eyetracker gaze data”. In: Jan. 2010, pp. 267–270. DOI: 10.1145/1743666.1743729 (cit. on p. 19).

[PLX+19] D. Parikh, Y. Lu, Y. Xin, D. Wu, J. Pelz, G. Lu. “Where Am I Looking: Localizing Gaze In Reconstructed 3D Space”. In: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE. 2019, pp. 1–5 (cit. on p. 17).

[RD05] D. C. Richardson, R. Dale. “Looking to understand: The coupling between speakers’ and listeners’ eye movements and its relationship to discourse comprehension”. In: *Cognitive science* 29.6 (2005), pp. 1045–1060 (cit. on p. 14).

[RHOL13a] G. Ristovski, M. Hunter, B. Olk, L. Linsen. “EyeC: Coordinated Views for Interactive Visual Exploration of Eye-Tracking Data”. In: July 2013, pp. 239–248. DOI: 10.1109/IV.2013.32 (cit. on p. 20).

[RHOL13b] G. Ristovski, M. Hunter, B. Olk, L. Linsen. “EyeC: Coordinated views for interactive visual exploration of eye-tracking data”. In: *2013 17th International Conference on Information Visualisation*. IEEE. 2013, pp. 239–248 (cit. on pp. 14, 15).

[RHOL13c] G. Ristovski, M. Hunter, B. Olk, L. Linsen. “EyeC: Coordinated views for interactive visual exploration of eye-tracking data”. In: *2013 17th International Conference on Information Visualisation*. IEEE. 2013, pp. 239–248 (cit. on p. 15).

[RKT22] M. Ruotsalainen, L. Komulainen, A. Tejada. “Virtual Lenses to Support Study Work at Customer Site”. In: Jan. 2022. DOI: 10.13140/RG.2.2.29136.43521 (cit. on p. 25).

[ST98] R. Spence, L. Tweedie. “The Attribute Explorer: information synthesis via exploration”. In: *Interacting with Computers* 11.2 (1998), pp. 137–146 (cit. on p. 40).

[Sut68] I. E. Sutherland. “A head-mounted three dimensional display”. In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. 1968, pp. 757–764 (cit. on p. 13).

[TTS10a] H. Y. Tsang, M. Tory, C. Swindells. “eSeeTrack—visualizing sequential fixation patterns”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 953–962 (cit. on p. 14).

[TTS10b] H. Y. Tsang, M. Tory, C. Swindells. “eSeeTrack—visualizing sequential fixation patterns”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 953–962 (cit. on p. 15).

[UBG+20] D. Ungureanu, F. Bogo, S. Galliani, P. Sama, X. Duan, C. Meekhof, J. Stühmer, T. J. Cashman, B. Tekin, J. L. Schönberger, et al. “Hololens 2 research mode as a tool for computer vision research”. In: *arXiv preprint arXiv:2008.11239* (2020) (cit. on pp. 25, 26).

[WFE+12] N. Weibel, A. Fouse, C. Emmenegger, S. Kimmich, E. Hutchins. “Let’s look at the cockpit: exploring mobile eye-tracking for observational research on the flight deck”. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*. 2012, pp. 107–114 (cit. on p. 14).

[WHRK06] J. M. West, A. R. Haake, E. P. Rozanski, K. S. Karn. “eyePatterns: software for identifying patterns and similarities across fixation sequences”. In: *Proceedings of the 2006 symposium on Eye tracking research & applications*. 2006, pp. 149–154 (cit. on p. 15).

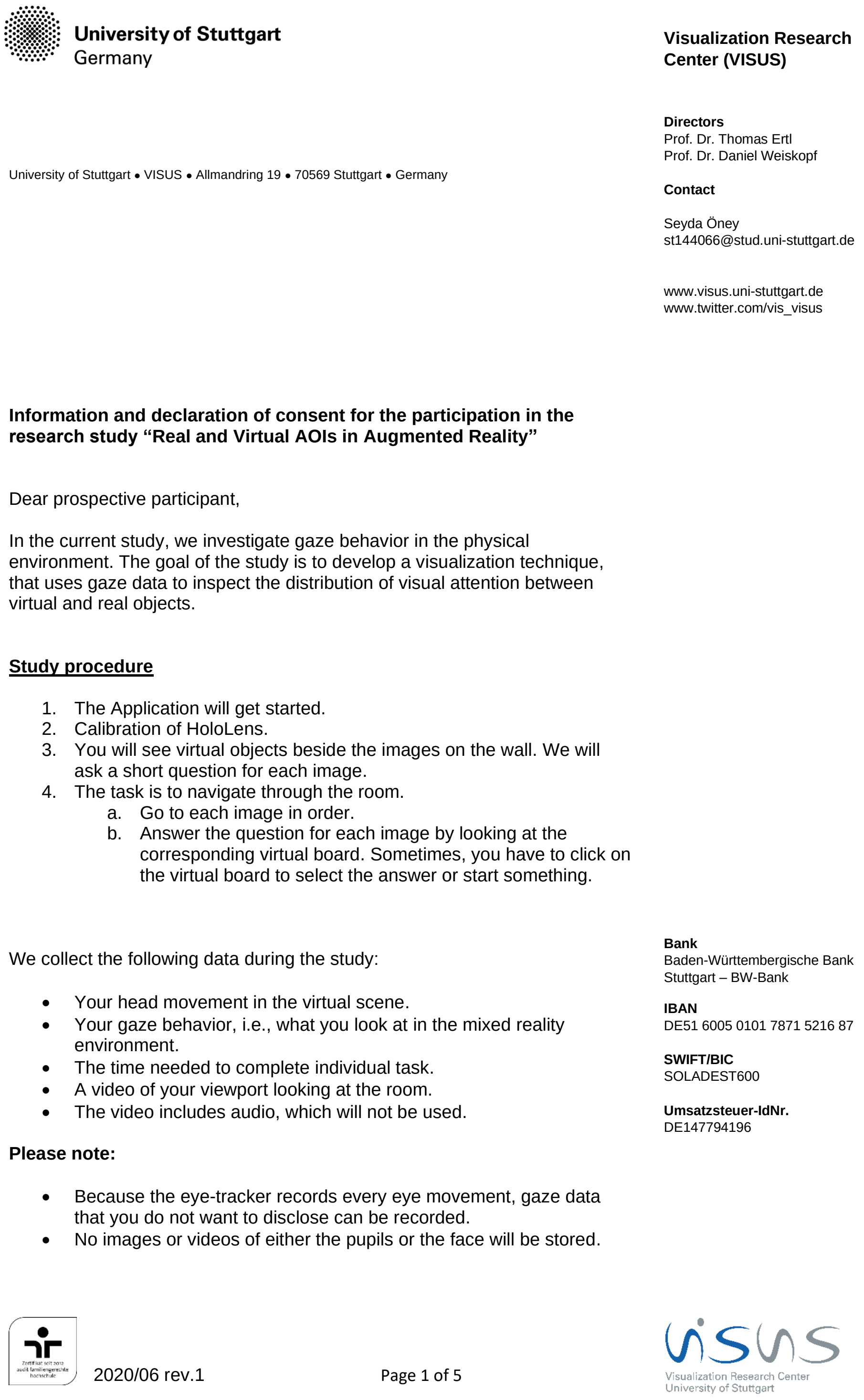
[YKSJ07] J. S. Yi, Y. ah Kang, J. Stasko, J. A. Jacko. “Toward a deeper understanding of the role of interaction in information visualization”. In: *IEEE transactions on visualization and computer graphics* 13.6 (2007), pp. 1224–1231 (cit. on p. 37).

[YT21] M. Yan, M. Tamke. “Augmented Reality for Experience-centered Spatial Design-A quantitative assessment method for architectural space”. In: (2021) (cit. on p. 17).

All links were last followed on June 10, 2022.

A Appendix

A.1 Case Study



University of Stuttgart
Germany

Visualization Research Center (VISUS)

Directors
Prof. Dr. Thomas Ertl
Prof. Dr. Daniel Weiskopf

Contact

Seyda Öney
st144066@stud.uni-stuttgart.de

www.visus.uni-stuttgart.de
www.twitter.com/vis_visus

University of Stuttgart • VISUS • Allmandring 19 • 70569 Stuttgart • Germany

Information and declaration of consent for the participation in the research study “Real and Virtual AOIs in Augmented Reality”

Dear prospective participant,

In the current study, we investigate gaze behavior in the physical environment. The goal of the study is to develop a visualization technique, that uses gaze data to inspect the distribution of visual attention between virtual and real objects.

<u>Study procedure</u>

1. The Application will get started.
2. Calibration of HoloLens.
3. You will see virtual objects beside the images on the wall. We will ask a short question for each image.
4. The task is to navigate through the room.
   a. Go to each image in order.
   b. Answer the question for each image by looking at the corresponding virtual board. Sometimes, you have to click on the virtual board to select the answer or start something.

We collect the following data during the study:

- Your head movement in the virtual scene.
- Your gaze behavior, i.e., what you look at in the mixed reality environment.
- The time needed to complete individual task.
- A video of your viewport looking at the room.
- The video includes audio, which will not be used.

Please note:

- Because the eye-tracker records every eye movement, gaze data that you do not want to disclose can be recorded.
- No images or videos of either the pupils or the face will be stored.

Bank
Baden-Württembergische Bank
Stuttgart – BW-Bank

IBAN
DE51 6005 0101 7871 5216 87

SWIFT/BIC
SOLADEST600

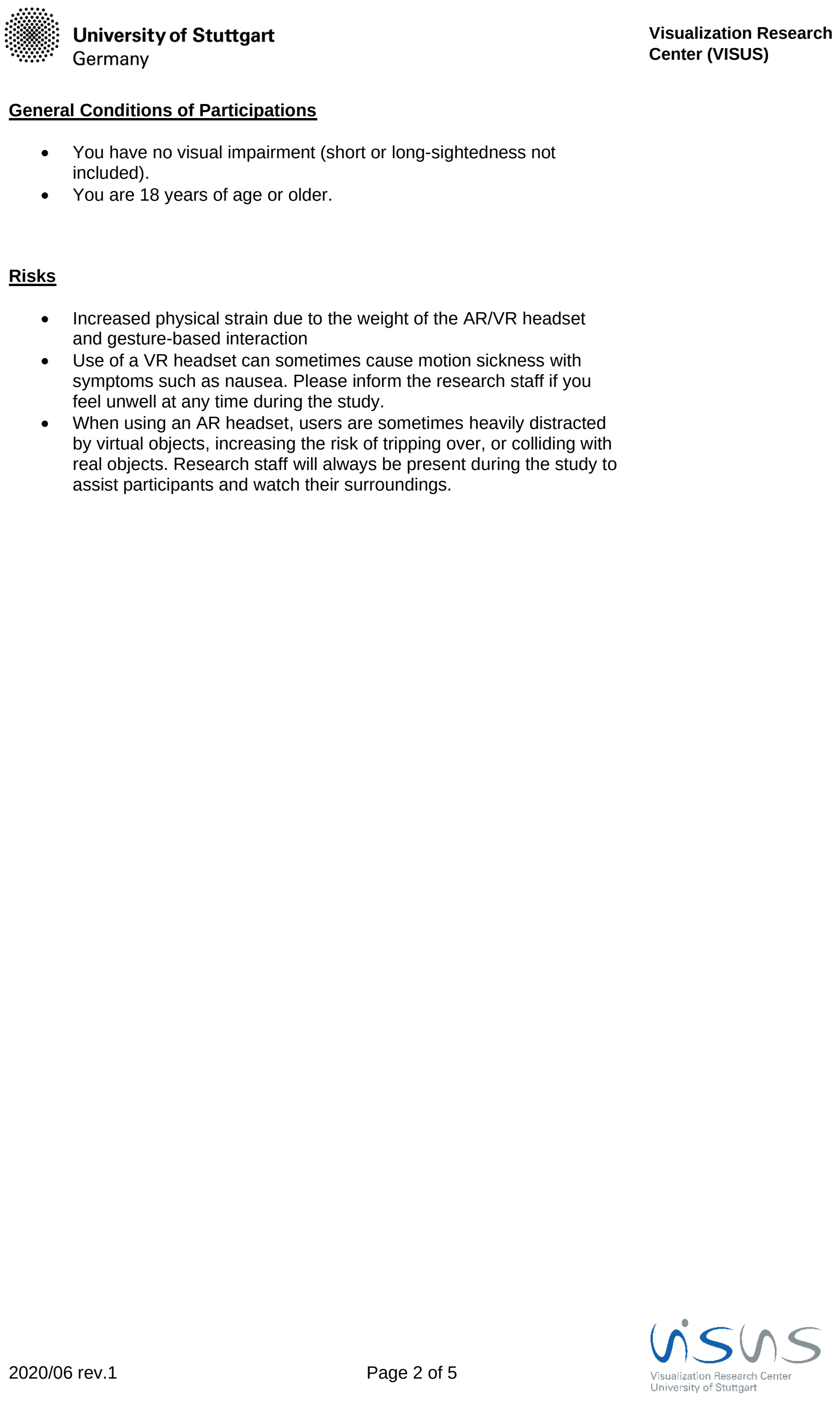
Umsatzsteuer-IdNr.
DE147794196

2020/06 rev.1

Page 1 of 5

Visualization Research Center
University of Stuttgart

Figure A.1: consent form



University of Stuttgart
Germany

Visualization Research Center (VISUS)

General Conditions of Participations

- You have no visual impairment (short or long-sightedness not included).
- You are 18 years of age or older.

Risks

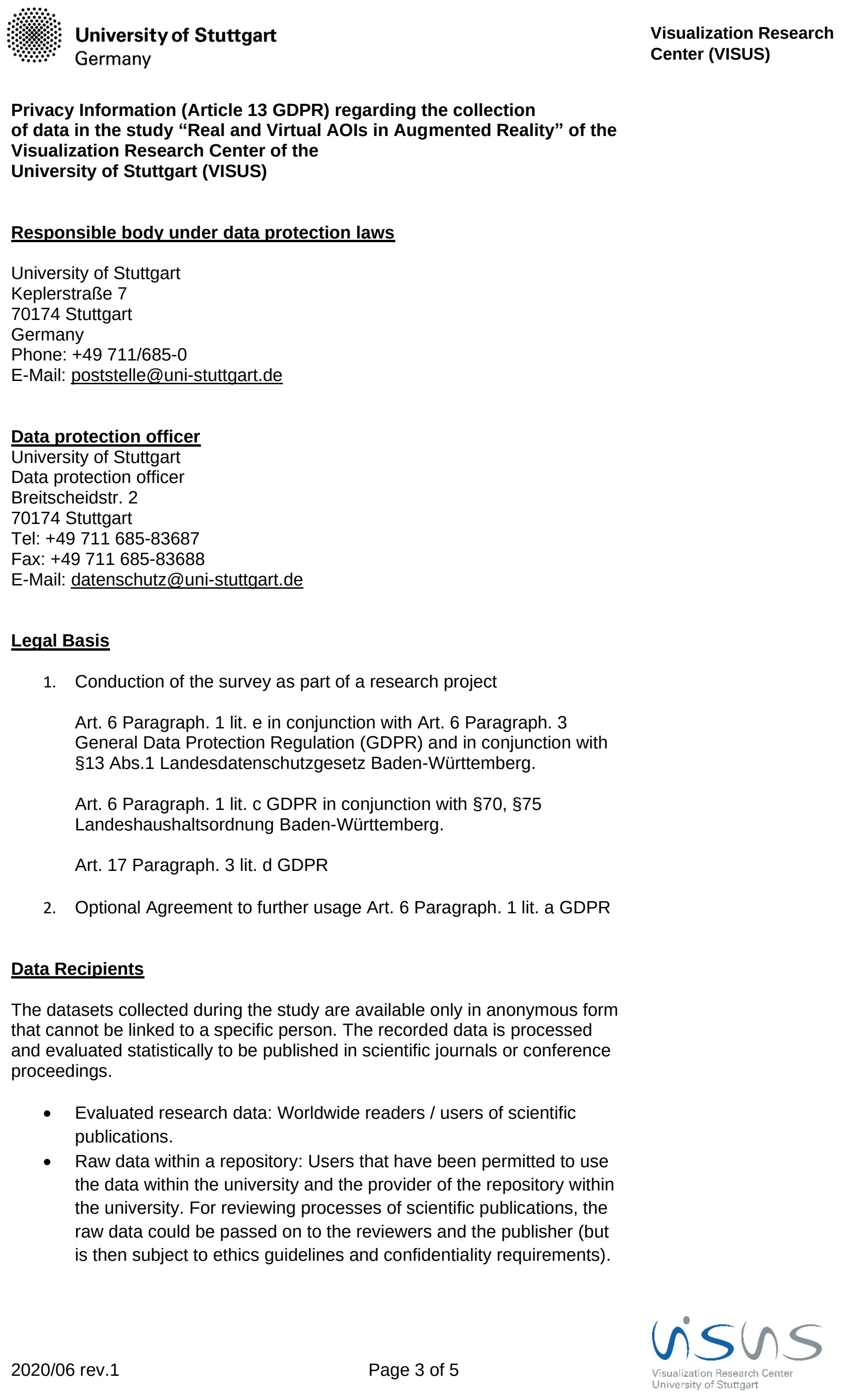
- Increased physical strain due to the weight of the AR/VR headset and gesture-based interaction
- Use of a VR headset can sometimes cause motion sickness with symptoms such as nausea. Please inform the research staff if you feel unwell at any time during the study.
- When using an AR headset, users are sometimes heavily distracted by virtual objects, increasing the risk of tripping over, or colliding with real objects. Research staff will always be present during the study to assist participants and watch their surroundings.

2020/06 rev.1

Page 2 of 5

Visualization Research Center
University of Stuttgart

Figure A.2: consent form



University of Stuttgart
Germany

Visualization Research
Center (VISUS)

Privacy Information (Article 13 GDPR) regarding the collection of data in the study “Real and Virtual AOIs in Augmented Reality” of the Visualization Research Center of the University of Stuttgart (VISUS)

Responsible body under data protection laws

University of Stuttgart
Keplerstraße 7
70174 Stuttgart
Germany
Phone: +49 711/685-0
E-Mail: poststelle@uni-stuttgart.de

Data protection officer
University of Stuttgart
Data protection officer
Breitscheidstr. 2
70174 Stuttgart
Tel: +49 711 685-83687
Fax: +49 711 685-83688
E-Mail: datenschutz@uni-stuttgart.de

Legal Basis

1. Conduction of the survey as part of a research project

   Art. 6 Paragraph. 1 lit. e in conjunction with Art. 6 Paragraph. 3 General Data Protection Regulation (GDPR) and in conjunction with §13 Abs.1 Landesdatenschutzgesetz Baden-Württemberg.

   Art. 6 Paragraph. 1 lit. c GDPR in conjunction with §70, §75 Landeshaushaltsordnung Baden-Württemberg.

   Art. 17 Paragraph. 3 lit. d GDPR

2. Optional Agreement to further usage Art. 6 Paragraph. 1 lit. a GDPR

Data Recipients

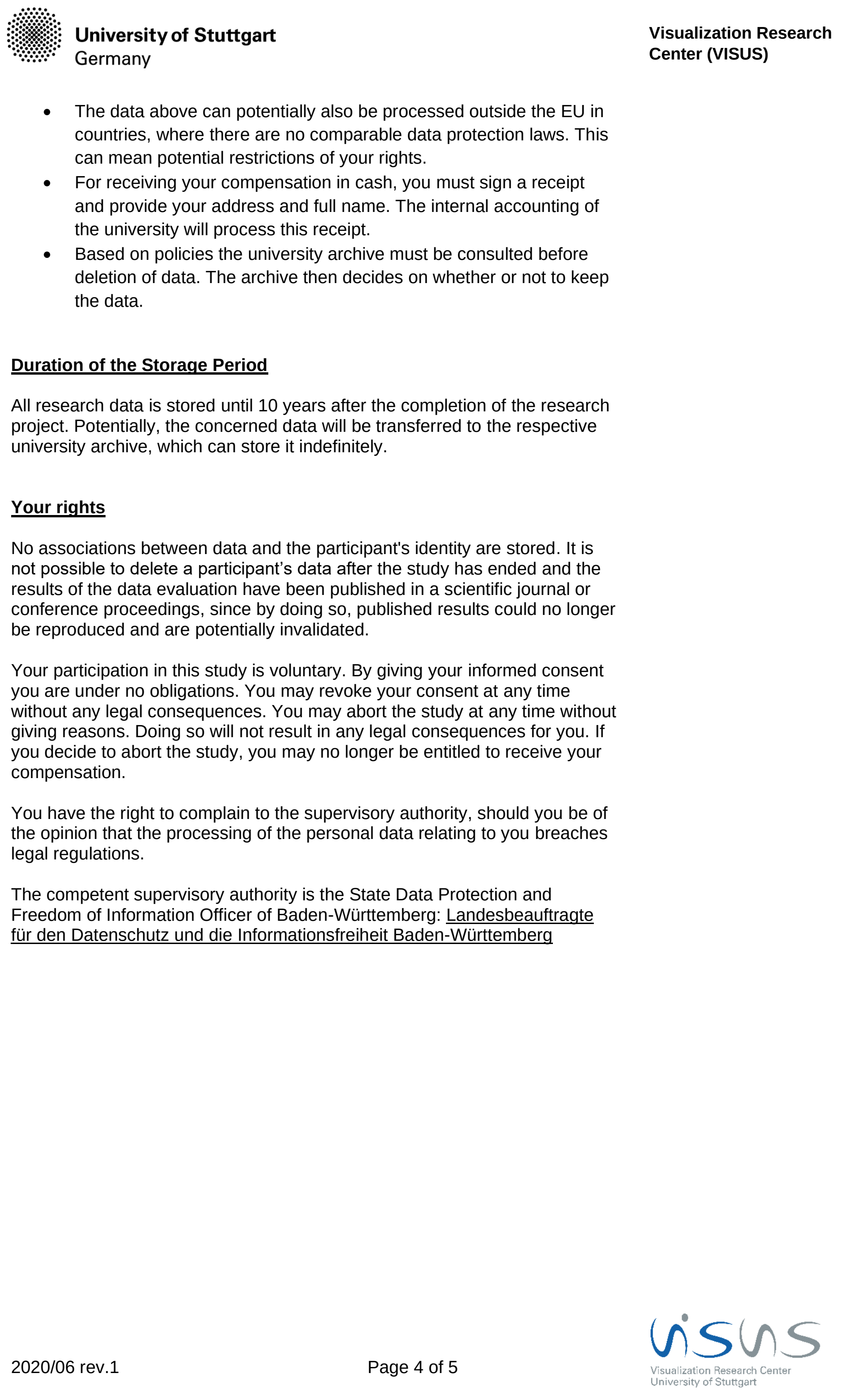
The datasets collected during the study are available only in anonymous form that cannot be linked to a specific person. The recorded data is processed and evaluated statistically to be published in scientific journals or conference proceedings.

- Evaluated research data: Worldwide readers / users of scientific publications.
- Raw data within a repository: Users that have been permitted to use the data within the university and the provider of the repository within the university. For reviewing processes of scientific publications, the raw data could be passed on to the reviewers and the publisher (but is then subject to ethics guidelines and confidentiality requirements).

2020/06 rev.1 Page 3 of 5

Visualization Research Center
University of Stuttgart

Figure A.3: consent form



University of Stuttgart
Germany

Visualization Research
Center (VISUS)

- The data above can potentially also be processed outside the EU in countries, where there are no comparable data protection laws. This can mean potential restrictions of your rights.
- For receiving your compensation in cash, you must sign a receipt and provide your address and full name. The internal accounting of the university will process this receipt.
- Based on policies the university archive must be consulted before deletion of data. The archive then decides on whether or not to keep the data.

Duration of the Storage Period

All research data is stored until 10 years after the completion of the research project. Potentially, the concerned data will be transferred to the respective university archive, which can store it indefinitely.

Your rights

No associations between data and the participant's identity are stored. It is not possible to delete a participant’s data after the study has ended and the results of the data evaluation have been published in a scientific journal or conference proceedings, since by doing so, published results could no longer be reproduced and are potentially invalidated.

Your participation in this study is voluntary. By giving your informed consent you are under no obligations. You may revoke your consent at any time without any legal consequences. You may abort the study at any time without giving reasons. Doing so will not result in any legal consequences for you. If you decide to abort the study, you may no longer be entitled to receive your compensation.

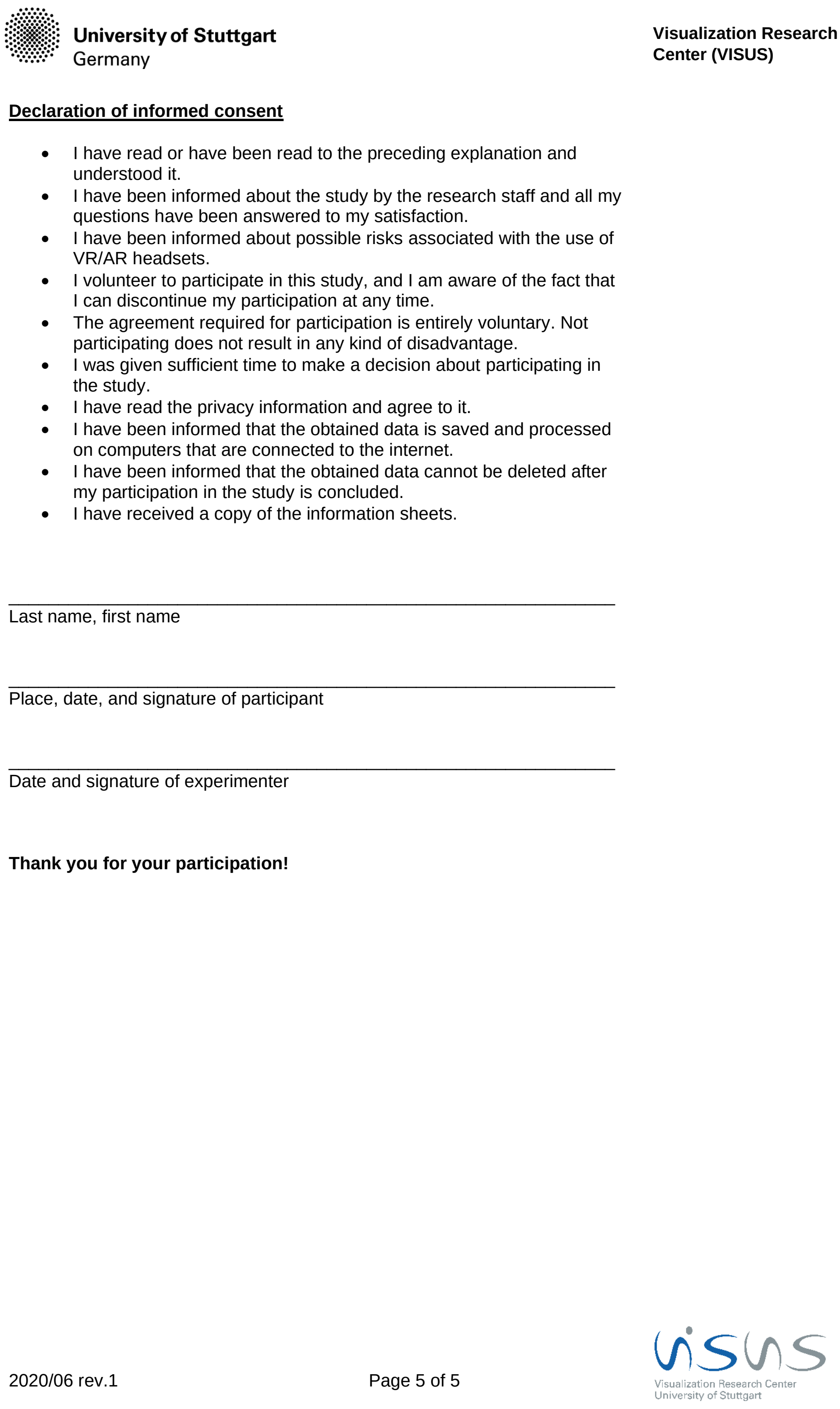
You have the right to complain to the supervisory authority, should you be of the opinion that the processing of the personal data relating to you breaches legal regulations.

The competent supervisory authority is the State Data Protection and Freedom of Information Officer of Baden-Württemberg: Landesbeauftragte für den Datenschutz und die Informationsfreiheit Baden-Württemberg

2020/06 rev.1 Page 4 of 5

Visualization Research Center
University of Stuttgart

Figure A.4: consent form



University of Stuttgart
Germany

Visualization Research Center (VISUS)

Declaration of informed consent

- I have read or have been read to the preceding explanation and understood it.
- I have been informed about the study by the research staff and all my questions have been answered to my satisfaction.
- I have been informed about possible risks associated with the use of VR/AR headsets.
- I volunteer to participate in this study, and I am aware of the fact that I can discontinue my participation at any time.
- The agreement required for participation is entirely voluntary. Not participating does not result in any kind of disadvantage.
- I was given sufficient time to make a decision about participating in the study.
- I have read the privacy information and agree to it.
- I have been informed that the obtained data is saved and processed on computers that are connected to the internet.
- I have been informed that the obtained data cannot be deleted after my participation in the study is concluded.
- I have received a copy of the information sheets.

____
Last name, first name

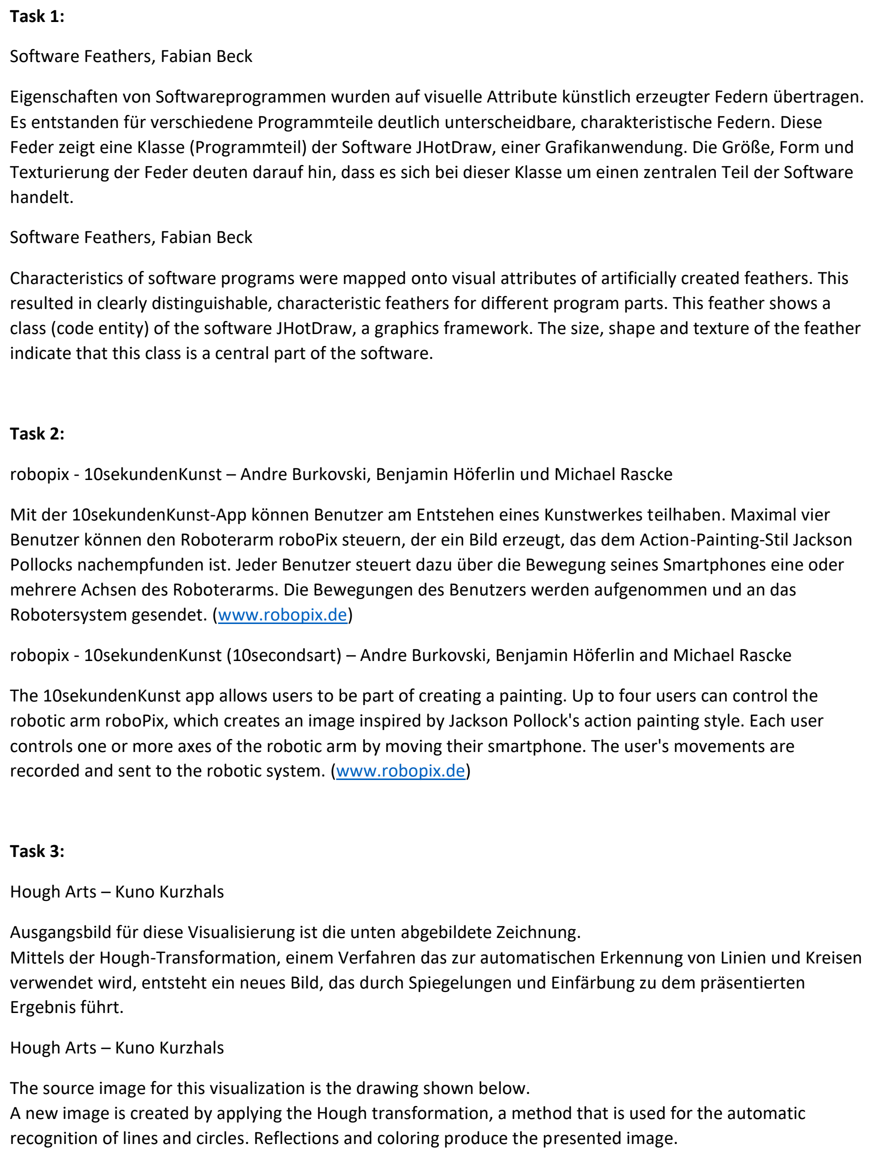
____
Place, date, and signature of participant

____
Date and signature of experimenter

Thank you for your participation!

2020/06 rev.1 Page 5 of 5

Figure A.5: consent form



Task 1:

Software Feathers, Fabian Beck

Eigenschaften von Softwareprogrammen wurden auf visuelle Attribute künstlich erzeugter Federn übertragen. Es entstanden für verschiedene Programmteile deutlich unterscheidbare, charakteristische Federn. Diese Feder zeigt eine Klasse (Programmteil) der Software JHotDraw, einer Grafikanwendung. Die Größe, Form und Texturierung der Feder deuten darauf hin, dass es sich bei dieser Klasse um einen zentralen Teil der Software handelt.

Software Feathers, Fabian Beck

Characteristics of software programs were mapped onto visual attributes of artificially created feathers. This resulted in clearly distinguishable, characteristic feathers for different program parts. This feather shows a class (code entity) of the software JHotDraw, a graphics framework. The size, shape and texture of the feather indicate that this class is a central part of the software.

Task 2:

robopix - 10sekundenKunst – Andre Burkovski, Benjamin Höferlin und Michael Rascke

Mit der 10sekundenKunst-App können Benutzer am Entstehen eines Kunstwerkes teilhaben. Maximal vier Benutzer können den Roboterarm roboPix steuern, der ein Bild erzeugt, das dem Action-Painting-Stil Jackson Pollocks nachempfunden ist. Jeder Benutzer steuert dazu über die Bewegung seines Smartphones eine oder mehrere Achsen des Roboterarms. Die Bewegungen des Benutzers werden aufgenommen und an das Robotersystem gesendet. (www.robopix.de)

robopix - 10sekundenKunst (10secondsart) – Andre Burkovski, Benjamin Höferlin and Michael Rascke

The 10sekundenKunst app allows users to be part of creating a painting. Up to four users can control the robotic arm roboPix, which creates an image inspired by Jackson Pollock's action painting style. Each user controls one or more axes of the robotic arm by moving their smartphone. The user's movements are recorded and sent to the robotic system. (www.robopix.de)

Task 3:

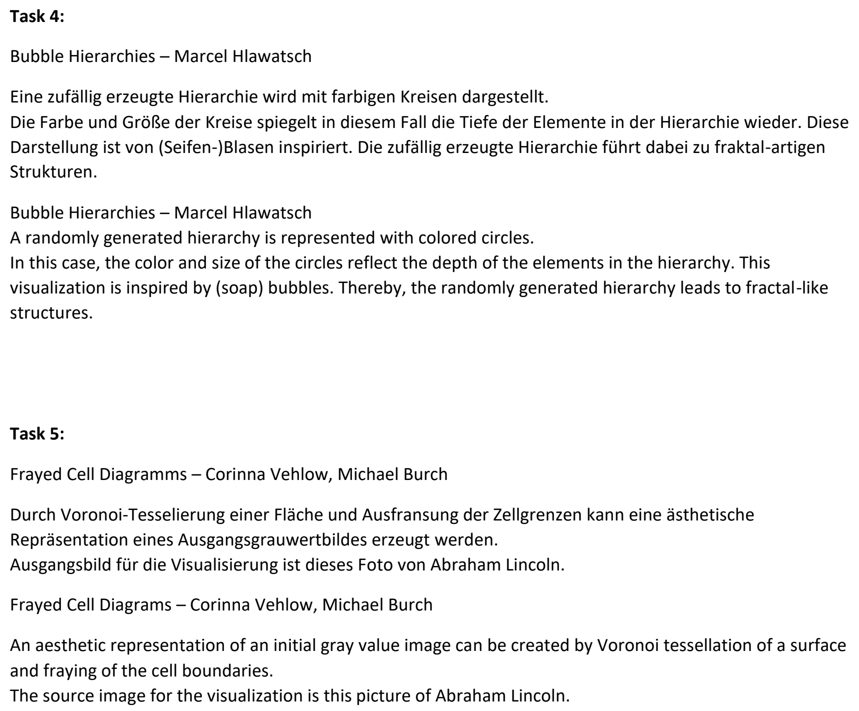
Hough Arts – Kuno Kurzhals

Ausgangsbild für diese Visualisierung ist die unten abgebildete Zeichnung.
Mittels der Hough-Transformation, einem Verfahren das zur automatischen Erkennung von Linien und Kreisen verwendet wird, entsteht ein neues Bild, das durch Spiegelungen und Einfärbung zu dem präsentierten Ergebnis führt.

Hough Arts – Kuno Kurzhals

The source image for this visualization is the drawing shown below.
A new image is created by applying the Hough transformation, a method that is used for the automatic recognition of lines and circles. Reflections and coloring produce the presented image.

Figure A.6: The descriptions of the images in tasks 1, 2 and 3 , which were read on the virtual boards, are shown.



Task 4:

Bubble Hierarchies – Marcel Hlawatsch

Eine zufällig erzeugte Hierarchie wird mit farbigen Kreisen dargestellt.
Die Farbe und Größe der Kreise spiegelt in diesem Fall die Tiefe der Elemente in der Hierarchie wieder. Diese Darstellung ist von (Seifen-)Blasen inspiriert. Die zufällig erzeugte Hierarchie führt dabei zu fraktal-artigen Strukturen.

Bubble Hierarchies – Marcel Hlawatsch
A randomly generated hierarchy is represented with colored circles.
In this case, the color and size of the circles reflect the depth of the elements in the hierarchy. This visualization is inspired by (soap) bubbles. Thereby, the randomly generated hierarchy leads to fractal-like structures.

Task 5:

Frayed Cell Diagramms – Corinna Vehlow, Michael Burch

Durch Voronoi-Tesselierung einer Fläche und Ausfransung der Zellgrenzen kann eine ästhetische Repräsentation eines Ausgangsgrauwertbildes erzeugt werden.
Ausgangsbild für die Visualisierung ist dieses Foto von Abraham Lincoln.

Frayed Cell Diagrams – Corinna Vehlow, Michael Burch

An aesthetic representation of an initial gray value image can be created by Voronoi tessellation of a surface and fraying of the cell boundaries.
The source image for the visualization is this picture of Abraham Lincoln.

Figure A.7: The descriptions of the images in tasks 4 and 5, which were read on the virtual boards, are shown.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature